

VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA  
EKONOMICKÁ FAKULTA

KATEDRA MATEMATICKÝCH METOD V EKONOMICE

Expertní systém pro úřady práce  
Expert System for Employment Agency

Student: RNDr. Karel Kíza  
Vedoucí diplomové práce: Ing. Václav Friedrich, Ph.D.

Ostrava 2011

## Zadání diplomové práce

Student: **RNDr. Karel Kíza**  
Studijní program: N6207 Kvantitativní metody v ekonomice  
Studijní obor: 6207T018 Kvantitativní podpora managementu  
Téma: **Expertní systém pro úřady práce**  
**Expert System for Employment Agency**

Zásady pro vypracování:

1. Úvod
  2. Teorie rozhodovacího procesu
  3. Použité metody a technologie
  4. Analýza a návrh expertního systému
  5. Závěr
- Seznam použité literatury  
Seznam zkratk  
Prohlášení o využití výsledků diplomové práce  
Přílohy

Seznam doporučené odborné literatury:


BRACHMAN, Donald J.; LEVESQUE, Hector J. *Knowledge Representation and Reasoning*. 1st ed. San Francisco: Morgan Kaufman Publisher (Elsevier), 2004. 381. s. ISBN 978-1-55860-932-7.  
FOTR, Jiří; ŠVECOVÁ, Lenka a kol. *Manažerské rozhodování. Postupy, metody a nástroje*. 1. vyd. Praha: Ekopress, 2010. 474 s. ISBN 978-80-86929-59-0.  
MERRITT, Denis. *Building Expert Systems in Prolog*. 1st ed. New York: Springer, 1989. 358 s. ISBN 978-03-87970-16-5.


Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Václav Friedrich, Ph.D.**

Datum zadání: 26.11.2010  
Datum odevzdání: 29.04.2011



  
doc. RNDr. Dana Šalounová, Ph.D.  
vedoucí katedry

  
prof. Dr. Ing. Dana Dluhošová  
děkanka fakulty

## **Prohlášení**

Prohlašuji, že tato diplomová práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Dne .....

.....  
jméno a příjmení studenta

## **Poděkování**

Zde bych chtěl poděkovat všem, kteří mi při zpracování diplomové práce pomáhali. Zejména vedoucímu mé diplomové práce Ing. Václavu Friedrichovi, Ph.D. za jeho zájem, věcné připomínky a čas, který věnoval mé práci.

# Obsah

1.	Úvod.....	8
1.1.	Nástin řešené problematiky.....	8
1.2.	Cíle práce .....	9
2.	Teorie rozhodování .....	11
2.1.	Rozhodovací problém .....	13
2.1.1.	Klasifikace rozhodovacích problémů .....	13
2.2.	Rozhodovací proces .....	14
2.2.1.	Prvky rozhodovacího procesu.....	14
2.2.1.1.	Cíl (cíle) rozhodování .....	14
2.2.1.2.	Kritéria hodnocení .....	15
2.2.1.3.	Subjekt rozhodování .....	15
2.2.1.4.	Objekt rozhodování.....	16
2.2.1.5.	Varianty rozhodování a jejich důsledky .....	16
2.2.1.6.	Stavy světa .....	16
2.2.2.	Členění rozhodovacích procesů podle informací o stavech světa a důsledcích variant .....	17
2.3.	Modely rozhodování .....	17
2.3.1.	Analytický model.....	17
2.3.2.	Racionálně ekonomický model.....	18
2.3.3.	Administrativní model .....	18
2.4.	Informace pro rozhodování.....	19
2.5.	Teorie výběru zaměstnání .....	19
2.5.1.	Výběr zaměstnání jako rozhodovací problém.....	19
2.5.1.1.	Identifikace a analýza rozhodovacího problému .....	20
2.5.1.2.	Stanovení kritérií pro hodnocení variant a tvorba variant řešení rozhodovacího problému .....	21
2.5.1.3.	Stanovení důsledků variant rozhodování .....	21
2.5.1.4.	Hodnocení důsledků variant rozhodování a výběr varianty určené k realizaci .....	21
2.5.1.5.	Realizace zvolené varianty rozhodování .....	22
2.5.1.6.	Kontrola výsledků realizované varianty .....	22
3.	Popis použitých metod a technologií .....	23
3.1.	Metody a nástroje vícekritériálního rozhodování .....	23
3.1.1.	Metody stanovení vah kritérií .....	23
3.1.1.1.	Metody přímého stanovení vah kritérií.....	24
3.1.1.2.	Metody stanovení vah kritérií založené na párovém srovnání.....	24
3.1.1.3.	Metoda postupného rozvrhu vah.....	26
3.1.1.4.	Kompenzační metoda stanovení vah .....	26
3.1.2.	Metody hodnocení variant .....	26
3.1.2.1.	Vícekritériální funkce utility.....	27
3.1.2.2.	Jednoduché metody stanovení užitku variant .....	28
3.1.2.3.	Metody založené na párovém srovnání variant .....	30
3.2.	Metody a technologie použité pro tvorbu expertního systémů .....	31
3.2.1.	Metodiky vývoje (expertních) systémů .....	31

3.2.1.1.	Tradiční metodiky vývoje software .....	32
3.2.2.	Expertní systémy .....	35
3.2.2.1.	Architektura expertních systémů .....	36
3.2.2.2.	Implementace expertních systémů .....	37
3.2.2.3.	Životní cyklus expertních systémů .....	37
3.2.2.4.	Výhody a nevýhody expertních systémů .....	38
4.	Analýza řešené problematiky a návrh expertního systému .....	39
4.1.	Analýza současného stavu .....	39
4.1.1.	Úřady práce – současná praxe .....	39
4.1.1.1.	Evidence a vyhledávání .....	39
4.1.1.2.	Dostupnost a reprezentace vstupních dat .....	40
4.1.2.	Hodnocení současné situace .....	43
4.1.2.1.	SWOT analýza současného stavu .....	43
4.1.2.2.	Příležitosti využitelné v navrhovaném systému .....	44
4.2.	Návrh rozhodovacího procesu .....	45
4.2.1.	Výběr kritérií pro vyhledávání .....	45
4.2.2.	Výběr metody stanovení vah kritérií .....	46
4.2.3.	Výběr metody hodnocení variant .....	46
4.2.4.	Výběr vývojového cyklu .....	46
4.3.	Návrh expertního systému .....	47
4.3.1.	Výběr hardwaru a softwaru .....	47
4.3.1.1.	Aplikační logika .....	47
4.3.1.1.1.	Programovací jazyk Java .....	48
4.3.1.1.2.	Programovací jazyk Prolog .....	49
4.3.1.2.	Zpracování dat .....	50
4.3.1.2.1.	JAXB .....	51
4.3.1.3.	Vývojové prostředí .....	51
4.3.2.	Návrh uživatelského rozhraní .....	52
4.3.2.1.	Specifikace hodnot a vah kritérií .....	52
4.3.2.2.	Zobrazení výsledků .....	54
4.3.2.3.	Celkový návrh .....	55
4.4.	Implementace expertního systému .....	56
4.4.1.	Prototyp .....	56
4.4.1.1.	Zpracování vstupních dat .....	56
4.4.1.2.	Transformace informací do znalostní báze .....	58
4.4.1.3.	Uživatelské rozhraní .....	59
4.4.1.4.	Hodnocení variant .....	59
4.4.1.5.	Prezentace výsledků .....	60
4.4.2.	Hodnocení prototypu a návrhy vylepšení .....	60
4.4.3.	Implementace výsledného systému .....	61
4.4.3.1.	Zpracování vstupních dat .....	61
4.4.3.2.	Uživatelské rozhraní .....	62
4.4.3.3.	Hodnocení variant .....	64
4.4.3.4.	Prezentace výsledků .....	66
4.4.3.5.	Použití výsledné aplikace .....	66
	Shrnutí a závěr .....	67

Literatura .....	68
Seznam zkratek .....	70
Prohlášení o využití výsledků diplomové práce .....	71
Seznam příloh na CD .....	72

# 1. Úvod

## 1.1. *Nástin řešené problematiky*

Rozhodování patří ke každodenním činnostem člověka. S touto činností se lze setkat ve všech oborech lidské činnosti. Kvalita a výsledky jednotlivých rozhodnutí pak ovlivňují ať přímo, nebo nepřímo celý náš život. Nekvalitní rozhodování může být příčinou mnoha neúspěchů. Rozhodování se může týkat například investic, nastavení ceny pro prodej výrobku, firemní strategie atd. nebo z pohledu jednotlivce například výběru zaměstnání.

Rozhodování je obecně vnímáno jako volba mezi různými způsoby chování. Může být rozděleno do dvou základních skupin, kterými jsou rozhodování manažerské a rozhodování osobní. Manažerské rozhodování je jádrem řízení organizací a je tedy podřízeno strategii celé organizace. V rámci osobního rozhodování dochází zpravidla k naplňování vlastního zájmu a rozhodnutí bývá realizováno jedním konkrétním rozhodovatelem. I když neexistuje přesný návod jak se v různých situacích rozhodovat, existuje celá řada metod, které mohou rozhodování usnadnit a pomohou vyvarovat se chybným rozhodnutím. Touto problematikou, tj. tvorbou metod pro podporu rozhodování a jejich následnou aplikací v praxi, se zabývá teorie rozhodování.

Každé rozhodnutí by mělo vycházet z kvalitních analýz postavených na důkladném prostudování dat, jež jsou momentálně k dispozici. Zdroje dat mohou být různé, například nejrozumnější statistiky, výroční zprávy, tisk atd. Schopnost čtení dat, spolehlivost dat, interpretace a následné použití těchto dat má přímý vliv na kvalitu vlastního rozhodnutí.

V současné době se pro podporu řešení rozhodovacích problémů velmi často využívá informačních technologií v podobě informačních systémů s využitím osobních počítačů. Tato podpora rozhodování (informační systémy na podporu rozhodování) může být obecně rozdělena do následujících skupin:

- Manažerský informační systém (Management Information System, dále jen MIS) je informační systém, který: „zpracovává neseříděné údaje z databází, dle požadavků



(dotazů) uživatele, za účelem zkvalitnění vedení organizace. Výsledky dotazů se zobrazují v grafech, tabulkách nebo sestavách (reportech). MIS tvoří lidé, zařízení a procedury na sběr, třídění, analýzu, interpretaci a distribuci včasných a přesných informací pro marketingové rozhodování. Posláním MIS je integrovat (shromažďovat) informace na jedno místo a především poskytovat je v použitelné formě“, viz [1].

- Systémy na podporu rozhodování (Decision Support System, dále jen DSS) jsou systémy, které: „pomáhají svým uživatelům – manažerům – při realizaci řídicích a rozhodovacích činností v podnikání. Uživatel tu může srovnávat dílčí výsledky řešení se svými představami a podle toho ovlivňovat další průběh řešení. Tyto systémy poskytují uživateli nabídky řešení a případně kladením dotazů usměrňují jeho postup. Tyto systémy ale nenahrazují rozhodovatele, jejich výsledkem tedy není rozhodnutí, pouze dávají rozhodovateli soubor variant, urychlují a zpřesňují propočty jejich důsledků a kvantifikují rizika.“, viz [2].
- Expertní systémy (Expert System, dále jen ES) jsou počítačové programy, které v podstatě simulují činnost experta při řešení problémů. Základ ES tvoří vhodně zakódované znalosti, které jsou převzaty od experta. Jejich cílem je simulovat experta a dosáhnout ve zvolené problémové oblasti srovnatelné kvality řešení problémů. Podpora poskytovaná těmito systémy může mít formu tvorby variant řešení, stanovení příčin daného problému apod. ES jako jediné informační systémy mohou v určitých případech rozhodovatele i nahradit.

## ***1.2. Cíle práce***

Jak již bylo nastíněno, rozhodování jako takové je oblastí, ve které existuje celá škála postupů pro řešení jednotlivých rozhodovacích problémů, ať už se jedná pouze o problém dílčí, nebo problém mající strategický význam. Tato práce bude zaměřena pouze na určitý výsek problematiky rozhodování, který bude sloužit k řešení konkrétního rozhodovacího problému. Nebylo by účelné pokoušet se zohlednit v práci všechny aspekty a přístupy v rozhodovací teorii existující.

Při dnešní úrovni nezaměstnanosti je otázka výběru zaměstnání, případně otázka výběru vhodných kandidátů pro určitou pracovní pozici velmi aktuální. Subjekty hledající vhodné

kandidáty (firmy) často svěřují výběr vhodných kandidátů na klíčové posty úřadům práce nebo specializovaným agenturám. Stejně tak i pro jednotlivce má výběr zaměstnání určující význam – pro jeho budoucnost, respektive pro jeho kariéru. Svěření výběru zaměstnání nebo vhodného kandidáta do rukou externí firmy (pracovního úřadu) je spojeno s rizikem jako každé delegování pravomoci. Může docházet ke zkreslení informací nebo nepochopení požadavků, ať už záměrnému nebo neúmyslnému. Tato skutečnost se může negativně promítnout v kvalitě procesu pro výběr zaměstnání (kandidáta) a tím ovlivnit i správnost rozhodnutí.

Tato práce bude zaměřena na problematiku výběru zaměstnání pro uchazeče o pracovní místo na pracovním úřadě. V současné době jsou nabídky práce evidovány jak na jednotlivých pracovních úřadech tak i v centrální databázi na Ministerstvu práce a sociálních věcí. Tato databáze umožňuje uchazečům prohlížení a vyhledávání pracovních nabídek. Neumožňuje však při vyhledávání zohlednit preference uživatelů, čímž by se užitek vyhledávání pro uživatele značně zvýšil.

Cílem této práce je vytvořit expertní systém usnadňující výběr pracovních míst pro uchazeče o práci (na pracovních úřadech), který zohlední při vyhledávání preference uživatelů a tím zefektivní výběr vhodných nabídek a v konečném důsledku i výběr kandidátů na jednotlivé pozice.

K dosažení tohoto hlavního cíle je třeba splnit následující dílčí cíle:

- a) provést analýzu současného stavu;
- b) navrhnout případná vylepšení oproti současnému stavu;

Na základě analýzy současného stavu:

- c) navrhnout rozhodovací proces pro hodnocení variant zohledňující preference uživatelů;
- d) navrhnout výsledný systém a zvolit vhodnou metodiku vývoje;
- e) s použitím vybrané metodiky implementovat navržený systém.

## 2. Teorie rozhodování

V této teoreticky zaměřené kapitole jsem vycházel především z prací Fotra [3] a Vebera [4]. Druhá kapitola věnovaná teorii rozhodování ve Vebrově knize [4] je v podstatě stejná jako ve Fotrově knize [3], čemuž odpovídá i fakt, že Fotr je uveden jako autor této kapitoly. Proto budu v rámci teorie rozhodování vycházet dále většinou z Fotrovy monografie [3].

Rozhodování je podle Fotra [3] popsáno jako proces volby mezi přesně definovanými racionálními variantami. Mělo by se jednat o dynamický vědomý proces výběru jedné z možných alternativ, kterou lze dosáhnout požadovaného cíle. Rozhodování je vyvrcholením duševních pochodů, které vedou k závěrům nebo rozhodnutím.

Rozhodování, resp. rozhodovací procesy, mají obecně dvě stránky:

- stránku meritorní (věcnou, obsahovou);
- stránku formálně-logickou (procedurální).

Meritorní stránka odráží odlišnosti jednotlivých rozhodovacích procesů, resp. jejich typů. Každý typ rozhodovacího procesu má své specifické rysy, které jsou zdrojem odlišností těchto procesů. Stránka formálně-logická je odrazem skutečnosti, že jednotlivé rozhodovací procesy mají určité společné rysy a vlastnosti, a to bez ohledu na jejich odlišnou obsahovou náplň. Teorie rozhodování se zabývá právě společnými rysy rozhodovacích procesů.

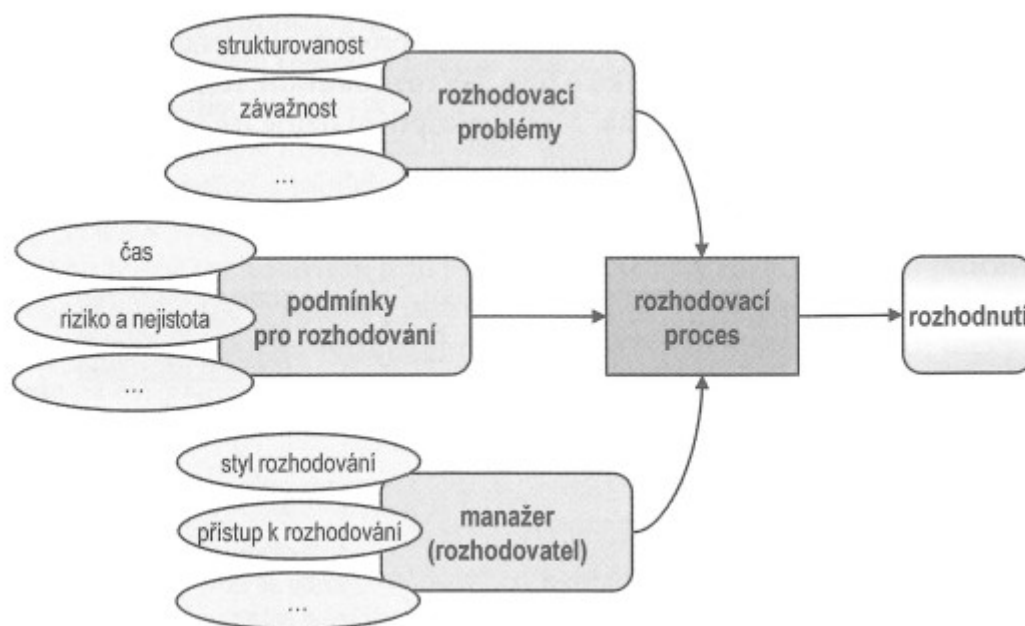
V průběhu historického vývoje došlo ke koncipování většího počtu teorií rozhodování, které se liší určitým způsobem pohledu na rozhodovací procesy, resp. soustředěním zájmu pouze na určité aspekty těchto procesů:

- teorie užitku (utility) – hodnocení variant podle kritérií;
- sociálně-psychologické teorie – zaměřené na subjekt a jeho chování;
- kvantitativně orientované – aplikace matematických modelů a metod (operační analýza, teorie her, rozhodovací analýza);
- teorie rozhodování v organizacích – respektuje omezené schopnosti subjektu rozhodování i omezení racionality v organizacích.

Z pohledu těchto teorií se v rámci této práce budeme zabývat teorií užitku, tj. hodnocení variant podle kritérií.

Odlišnosti jednotlivých teorií vyplývají též z jejich normativního, resp. deskriptivního charakteru.

- normativní teorie - poskytnutí návodů jak problémy řešit, jaké modely používat, jakým způsobem a tvorba norem, jejichž aplikace vede k žádoucí kvalitě rozhodování;
- deskriptivní teorie - získávání poznatků o tom, jak rozhodování skutečně probíhá, popis, analýza a hodnocení rozhodovacích procesů, jejich průběhu, předností a nedostatků, chování rozhodovatele a dalších subjektů v průběhu rozhodovacího procesu.



Obr. 2.1. Struktura rozhodovacího procesu, viz Fotr [3].

Jak je vidět z obrázku 2.1, je výsledné rozhodnutí založeno na třech základních vstupech, a to:

- rozhodovacím problému,
- podmínkách pro rozhodnutí
- a manažerovi (rozhodovateli).

V literatuře bývají pojmy rozhodovací proces a rozhodovací problém často zaměňovány, nebo jsou používány jako synonyma. Fotr [3] však tyto pojmy rozlišuje, a proto, nežli přistoupíme k definici rozhodovacího procesu, budeme nejprve definovat rozhodovací problém.

## ***2.1. Rozhodovací problém***

Fotr [3] definuje rozhodovací problém jako existenci odchylky mezi stavem žádoucím, tj. tím, co má být, např. podle standardu, normy, či plánu, a stavem skutečným, tj. tím, co se skutečně stalo. Dle Vebera [4] je rozhodovací problém první etapou rozhodovacího procesu a k jeho řešení nelze přistoupit dříve, než bude problém identifikován.

### **2.1.1. Klasifikace rozhodovacích problémů**

Problémy se obecně rozdělují na:

- potenciální problémy, které mohou vzniknout v budoucnu;
- problémy reálné, které již existují a mohou se lišit svou naléhavostí, rozsahem a možnými dopady v případě jejich neřešení.

Z pohledu tohoto členění se bude tato práce zabývat problémem reálným, tj. skutečně existujícím.

Další způsob členění rozhodovacích problémů je z hlediska jejich složitosti a možnosti algoritmizace, a to na:

- Dobře strukturované rozhodovací problémy – jsou také označovány jako problémy jednoduché, programované, algoritmizované. Zpravidla se opakovaně řeší na operativní úrovni a existují pro ně již známé rutinní postupy. Jsou charakterizovány kvantifikovatelnými proměnnými s jediným kvantitativním kritériem hodnocení.
- Špatně strukturované rozhodovací problémy – jsou řešeny především na vyšších úrovních řízení, svým charakterem jsou vždy do určité míry nové a neopakovatelné, při jejich řešení je třeba tvůrčí přístup a zkušenosti, přičemž neexistují žádné standardní procedury jejich řešení. Existuje větší počet kritérií a variant řešení a zpravidla je obtížná interpretace informací potřebných pro rozhodnutí.

Z pohledu členění rozhodovacích problémů dle složitosti a možnosti algoritmizace se tato práce bude zabývat problémem dobře strukturovaným.

## **2.2. Rozhodovací proces**

Rozhodovací proces lze chápat jako proces řešení rozhodovacího problému s více (tj. alespoň dvěma) variantami řešení. Jestliže vycházíme z toho, že základním atributem rozhodování je:

- proces volby, tj. posuzování jednotlivých variant
- a výběr rozhodnutí, tj. optimální varianty, resp. varianty určené k realizaci,

pak problémy s jediným řešením (tj. existuje pouze jediné řešení, resp. bylo nalezeno pouze jediné řešení) nejsou skutečnými rozhodovacími problémy a řešení těchto problémů tedy není rozhodovacím procesem ve smyslu jak jej chápe Fotr [3].

Podstatou rozhodovacího procesu je tedy volba alespoň mezi dvěma možnostmi, tj. dvěma variantami rozhodování.

Rozhodovací proces je řetězec určitých úkonů řazených za sebou s určitou logickou posloupností, odvíjející se a směřující od podnětu k rozhodnutí až po vydání rozhodnutí, jeho realizaci a následné kontrole.

### **2.2.1. Prvky rozhodovacího procesu**

Mezi základní stavební prvky rozhodovacího procesu patří:

- cíl rozhodování;
- kritéria hodnocení;
- subjekt a objekt rozhodování;
- varianty rozhodování a jejich důsledky;
- stavy světa.

#### **2.2.1.1. Cíl (cíle) rozhodování**

Cíl rozhodování (řešení rozhodovacího problému) lze chápat jako určitý stav, který je žádoucí a kterého má být dosaženo řešením rozhodovacího problému. U většiny rozhodovacích problémů existuje větší počet (dílčích) cílů, kterých má být dosaženo. Mezi jednotlivými dílčími cíli

mohou vznikat vzájemné vztahy, a to vztahy komplementární, kdy se cíle vzájemně doplňují a podporují, nebo konfliktní, kdy dosažení vysokých hodnot jednoho cíle je spojeno zpravidla s nízkými hodnotami jiného cíle. Důležitým faktorem řešení rozhodovacích problémů je forma jejich vyjádření. Ta je rozdělena na formu číselnou u cílů kvantitativní povahy a formu popisnou (slovní) u cílů kvalitativní povahy.

Cílem rozhodování je v našem případě vybrat pro uchazeče o práci co nejvhodnější (optimální) pracovní nabídku.

#### **2.2.1.2. Kritéria hodnocení**

Kritéria hodnocení představují hlediska zvolená rozhodovatelem (na základě jeho hodnotové soustavy) sloužící k posouzení výhodnosti jednotlivých variant rozhodování z hlediska dosažení, resp. stupně plnění dílčích cílů řešeného rozhodovacího problému. Kritéria jsou zpravidla odvozena od cílů řešení rozhodovacích problémů. Uplatnění kritérií hodnocení při posuzování výhodnosti jednotlivých variant vyžaduje chápat odlišnosti jednotlivých kritérií. Proto je třeba rozlišovat kritéria, jejichž důsledky variant vzhledem k těmto kritériím jsou vyjádřeny buď číselně (pro kvantitativní kritéria), nebo slovně (pro kvalitativní kritéria). Předností kvantitativních kritérií je jasná náplň, jednoznačný smysl pro rozhodovatele, snadná měřitelnost. Dále jsou kvantitativní kritéria rozlišována podle jejich způsobu naplňování cíle na kritéria nákladového typu a kritéria výnosového typu. U kritérií nákladového typu jsou preferovány nižší hodnoty před vyššími, u kritérií výnosového typu naopak hodnoty vyšší před nižšími.

Kritéria hodnocení budou v našem případě vycházet z vlastností (atributů) evidovaných u jednotlivých nabídek práce.

#### **2.2.1.3. Subjekt rozhodování**

Subjektem rozhodování (rozhodovatelem) je označována osoba, která volí ze skupiny možných variant tu, která bude realizována. Subjektem rozhodování může být:

- jednotlivec – hovoříme pak o individuálním subjektu rozhodování;
- skupina lidí (orgán), kdy je rozhodovatelem skupina osob.

Dále se v praxi subjekty rozhodování dělí na:

- statutární rozhodovatele – subjekt je vybaven pravomocemi k volbě varianty určené k realizaci a nese odpovědnost za dopady volby této varianty;
- skutečné rozhodovatele – subjekt, který skutečně rozhoduje.

Subjektem rozhodování je v rámci této práce jednotlivec (člověk hledající práci), jedná se o skutečného rozhodovatele.

#### **2.2.1.4. Objekt rozhodování**

Objektem rozhodování se chápe zpravidla oblast (organizační jednotka), které se rozhodování týká (v jejímž rámci se problém formuloval).

Objektem rozhodování je v našem případě výběr zaměstnání.

#### **2.2.1.5. Varianty rozhodování a jejich důsledky**

Jedná se v podstatě o jeden z možných způsobů jednání řešitele, které má vést k naplnění vymezených cílů, resp. k vyřešení daného problému. U některých rozhodovacích problémů jsou varianty rozhodovacích problémů dopředu známy, u jiných je tvorba variant složitým procesem vyhledávání a zpracování informací. S variantami řešení jsou spojeny důsledky, které jsou chápány jako předpokládané dopady (účinky) v důsledku volby určité varianty. Důsledky variant jsou vyjadřovány vždy vzhledem k jednotlivým kritériím hodnocení. U kvantitativních kritérií se vyjadřují důsledky variant vzhledem k danému kritériu jako hodnota kritéria. U kritérií kvalitativní povahy jsou důsledky variant vyjádřeny slovními popisy (hodnota kritéria nemá smysl) a jsou označovány jako důsledky varianty vzhledem k danému kritériu hodnocení.

Varianty rozhodování budou v našem případě odpovídat jednotlivým evidovaným pracovním nabídkám.

#### **2.2.1.6. Stavy světa**

Stavy světa nebo také stavy okolí či rizikové situace chápeme jako budoucí vzájemně se vylučující situace, které mohou po realizaci varianty nastat a tím ovlivňují důsledky této varianty



vzhledem k některým kritériím. Stavby světa hrají významnou roli v případě rozhodování za rizika (nejistoty).

### **2.2.2. Členění rozhodovacích procesů podle informací o stavech světa a důsledcích variant**

Rozhodovací procesy (rozhodování) členíme dle informací o stavech světa a důsledků variant na:

- rozhodování za jistoty – rozhodovatel ví s jistotou, který stav světa nastane a jaké budou důsledky variant (rozhodne-li se rozhodovatel pro řešení A, bude mít jeho jednání jediný důsledek B);
- rozhodování za rizika – rozhodovatel zná možné budoucí situace (stavy světa), které mohou nastat, a tím i důsledky jednotlivých variant, a současně zná pravděpodobnost těchto stavů světa (rozhodne-li se rozhodovatel pro řešení A, bude mít jeho jednání několik alternativ důsledků, např. B,C,D, u kterých jsou však známy pravděpodobnosti, se kterými tyto důsledky nastanou);
- rozhodování za nejistoty – rozhodovatel zná možné budoucí situace (stavy světa), ale nezná pravděpodobnosti, se kterými mohou nastat.

V našem případě bude důsledkem situace, kdy rozhodovatel zvolí určitou variantu (nabídku práce), jeho podání žádosti o toto místo. Z tohoto důvodu jde o vícekritériální rozhodování za jistoty.

## **2.3. Modely rozhodování**

### **2.3.1. Analytický model**

Jedná se o model, kdy je celý rozhodovací proces rozdělen do etap. Těchto dělení je možné v literatuře najít několik v závislosti na granularitě rozložení. Jako příklad agregovanějšího členění rozhodovacího procesu lze uvést Fotrův přístup [3], který rozlišuje následující etapy:

- identifikace rozhodovacích problémů – identifikace situací, které vyžadují řešení;
- analýza a formulace rozhodovacích problémů – výsledkem je formulace rozhodovacího problému;
- stanovení kritérií hodnocení variant – stanovení kritérií, podle kterých se budou jednotlivé varianty posuzovat;

- tvorba variant řešení rozhodovacího problému – hledání a tvorba (analýza) možných směrů činnosti, které vedou k řešení daného problému;
- stanovení důsledků variant rozhodování – zjištění předpokládaných dopadů jednotlivých variant;
- hodnocení důsledků variant rozhodování a výběr varianty určené k realizaci – určení buď optimální varianty, nebo preferenčního uspořádání variant;
- realizace zvolené varianty rozhodování – praktická implementace rozhodnutí;
- kontrola výsledků realizované varianty – stanovení odchylek dosažených výsledků vzhledem ke stanoveným cílům.

### **2.3.2. Racionálně ekonomický model**

Racionální rozhodování je chápáno jako rozhodování, které maximalizuje dosažení cílů, ať již jde o cíle jednotlivce, skupiny, nebo celých organizací. Ekonomicky racionální člověk se snaží systematicky vyhledávat nejlepší možná řešení problémů a tak maximalizovat svůj zisk. Mezi vlastnosti ekonomicky racionálního člověka patří:

- znalost všech variant vedoucí k dosažení stanoveného cíle;
- znalost důsledků těchto variant;
- schopnost kvantitativně ohodnotit každou variantu, tj. stanovit číselně její užitek;
- volba nejlepší varianty, tj. varianty s největším užitekem.

Postup racionálně ekonomického člověka při rozhodování začíná vyhledáním všech variant řešení daného problému, následuje jejich číselné ohodnocení z hlediska všech kritérií a nakonec zvolení nejvýhodnější varianty. Tento model je v souladu s analytickým modelem. Jeho specifická je v komplexní znalosti všech variant řešení, jejich dokonalém ohodnocení a volbě optimální varianty.

### **2.3.3. Administrativní model**

Na rozdíl od racionálně ekonomického modelu, který popisuje spíše rozhodování jednotlivce, popisuje administrativní model spíše rozhodování v organizacích, kdy jsou splněny následující předpoklady:

- znalost pouze omezeného rozsahu informací;
- omezená schopnost řešit rozhodovací problémy;

- nestanovují se všechny varianty vedoucí k dosažení cíle a všechny jejich důsledky;
- nehledá se optimální varianta, ale varianta, která je dostatečně dobrá.

Tato práce bude postavena na racionálně ekonomickém modelu, který vychází z analytického modelu. Hodnocení je fází analytického modelu, která v podstatě předchází závěrečnému rozhodnutí. V této fázi hodnotíme jednotlivé vytvořené varianty řešení prostřednictvím nástrojů a metod rozhodovací analýzy. Tato fáze je pro tuto práci stěžejní, proto jí bude věnována samostatná kapitola 3.1.

#### ***2.4. Informace pro rozhodování***

Informace mají pro rozhodovací proces klíčovou roli. Samotné rozhodovací procesy se někdy chápou jako procesy získávání a transformace vstupních informací na výstupní (což odpovídá definici informačního systému), zahrnující interpretaci těchto informací. Efektivnost získávání informací mohou oslabovat některé nedostatky jako irelevantní informace (nelze je použít pro řešení problémů), nesprávné nebo nepřesné informace, které neodpovídají skutečnosti, nejednoznačné, resp. konfliktní informace.

Informace potřebné pro tuto práci budou získány z databáze Ministerstva práce a sociálních věcí.

#### ***2.5. Teorie výběru zaměstnání***

Jak již bylo uvedeno dříve, bude tato práce zaměřena na teorii užitku, tj. hodnocení variant podle kritérií. Cílem rozhodování bude vybrat pro uchazeče o práci nejvhodnější pracovní nabídku. Kritéria hodnocení budou vycházet z parametrů evidovaných u jednotlivých nabídek práce. Subjektem rozhodování bude jednotlivec (člověk hledající práci). Objektem rozhodování bude problematika výběru zaměstnání. Řešený problém je reálný a dobře strukturovaný a jedná se o rozhodování za jistoty.

##### **2.5.1. Výběr zaměstnání jako rozhodovací problém**

Rozhodovací problém řešený v rámci této práce splňuje základní předpoklady racionálně ekonomického modelu (viz podkapitola 2.3.2):

- Znalost všech variant vedoucí k dosažení stanoveného cíle – jednotlivé varianty představují jednotlivé nabídky práce, které jsou v evidenci úřadu práce. Získávání a evidence těchto nabídek není předmětem práce.
- Znalost důsledků těchto variant – podle jednotlivých nabídek zná rozhodovatel důsledky – zná výši mzdy, pracovní dobu, místo výkonu práce a tím i vzdálenost od svého bydliště atd.
- Schopnost kvantitativně ohodnotit každou variantu, tj. stanovit číselně její užitek – každý rozhodovatel je schopen posoudit (ohodnotit) své preference pro jednotlivá kritéria, podle kterých hodnotí jednotlivé varianty. Aplikací těchto ohodnocených kritérií na jednotlivé varianty pak dostává hodnocení jednotlivých variant.
- Volba nejlepší varianty, tj. varianty s největším užitekem – na základě výsledného hodnocení jednotlivých variant volí rozhodovatel variantu s nejlepším hodnocením.

Vzhledem k tomu, že řešení výběru zaměstnání odpovídá racionálně ekonomickému modelu, který vychází z analytického modelu, bude dále uveden problém výběru zaměstnání rozdělený do etap odpovídajících tomuto modelu.

#### **2.5.1.1. Identifikace a analýza rozhodovacího problému**

Výsledkem etap identifikace a analýzy má být identifikace situace, která vyžaduje řešení, a následná formulace rozhodovacího problému včetně určení příčin vzniku problému, což odpovídá zadání a cílům práce.

V současné době jsou nabídky práce evidovány jak na jednotlivých pracovních úřadech, tak i v centrální databázi na Ministerstvu práce a sociálních věcí. Tato databáze umožňuje uchazečům prohlížení a vyhledávání pracovních nabídek. Neumožňuje však vyhledávání s ohledem na preference uživatelů, což by značně zvýšilo užitek výsledků vyhledávání.

Řešením tohoto problému je vytvořit systém (expertní) usnadňující výběr pracovních míst pro uchazeče o práci na pracovních úřadech, který zohlední při vyhledávání preference uživatelů a tím zefektivní výběr vhodných nabídek a v konečném důsledku i výběr kandidátů na jednotlivé pozice.

#### **2.5.1.2. Stanovení kritérií pro hodnocení variant a tvorba variant řešení rozhodovacího problému**

Jak uvádí Fotr [3], měly by tyto dvě etapy probíhat v úzké a vzájemné návaznosti. Důvodem je, že zvolená kritéria určují aspekty variant, které budou předmětem hodnocení a ovlivní tvorbu optimální varianty. Výsledkem těchto etap by měl být soubor kritérií, podle kterých se budou jednotlivé varianty posuzovat, a také soubor variant, které jsou možným řešením daného problému.

V tomto případě budou jak kritéria, tak jednotlivé varianty dány dostupným souborem kritérií, resp. variant, dostupným v centrální databázi na Ministerstvu práce a sociálních věcí. Soubor kritérií bude dán vlastnostmi evidovanými u nabídek práce (výše platu, obor, požadované vzdělání atd.) a jednotlivé varianty budou odpovídat jednotlivým nabídkám práce.

#### **2.5.1.3. Stanovení důsledků variant rozhodování**

Tato fáze těsně navazuje, respektive se prolíná s etapou tvorby variant. Významným pomocníkem při stanovení důsledků variant jsou experti, kteří jsou odborníky v dané oblasti a mohou tedy stanovit důsledky variant vzhledem k jednotlivým kritériím.

V tomto konkrétním případě bude v roli experta vystupovat sám uživatel (úředník na úřadě práce nebo uchazeč o zaměstnání) systému. Stanovení důsledků variant vzhledem k jednotlivým kritériím bude spočívat v (subjektivním) hodnocení jednotlivých variant z hlediska efektivity řešení výběru zaměstnání.

#### **2.5.1.4. Hodnocení důsledků variant rozhodování a výběr varianty určené k realizaci**

Cílem této fáze je stanovení takové varianty řešení problému, která nejlépe splňuje cíle řešení, tj. je nejlepší z hlediska celého souboru kritérií hodnocení.

V rámci výsledného systému budou jednotlivé varianty hodnoceny vzhledem k souboru kritérií a preferencím uživatelů. Výsledky budou zobrazeny uživateli, který získá seznam nejvhodnějších variant v daném pořadí.

Definice rozhodovacího problému splňuje předpoklady pro vícekritériální rozhodování, což jsou:

- multikriteriální charakter rozhodovacích problémů (varianty jsou hodnoceny z více hledisek (výše platu, obor, požadované vzdělání atd.);
- neaditivnost kritérií (kritéria jsou uváděna v různých měrných jednotkách (plat (Kč), dojezdnost (min) atd.);
- smíšený soubor kritérií (v souboru kritérií jsou zastoupena jak kritéria kvantitativní, tak i kvalitativní).

V tomto případě se tedy bude jednat o rozhodování při použití více kritérií, tj. vícekriteriální hodnocení. Vzhledem k tomu, že (vícekriteriální) hodnocení je stěžejní fází pro praktickou část této práce, bude metodám a nástrojům vícekriteriálního rozhodování věnována samostatná kapitola 3.1.

#### **2.5.1.5. Realizace zvolené varianty rozhodování**

Výsledkem této etapy je fyzická realizace (implementace) zvoleného řešení. V tomto případě se může jednat o:

- domluvení pracovního pohovoru v případě, že je uživatelem uchazeč o práci;
- nabídky pracovních příležitostí uchazeči v případě, že je uživatelem pracovník pracovního úřadu.

#### **2.5.1.6. Kontrola výsledků realizované varianty**

Kontrola výsledků je na uživateli (ať už se jedná o přímého uživatele, nebo pracovníka pracovního úřadu), tj. jeho hodnocení, nakolik vybraná pracovní nabídka odpovídá jeho požadavkům, případně možnostem. V případě neodpovídajících výsledků přichází do úvahy změna kritérií a preferencí pro vyhledávání.

### 3. Popis použitých metod a technologií

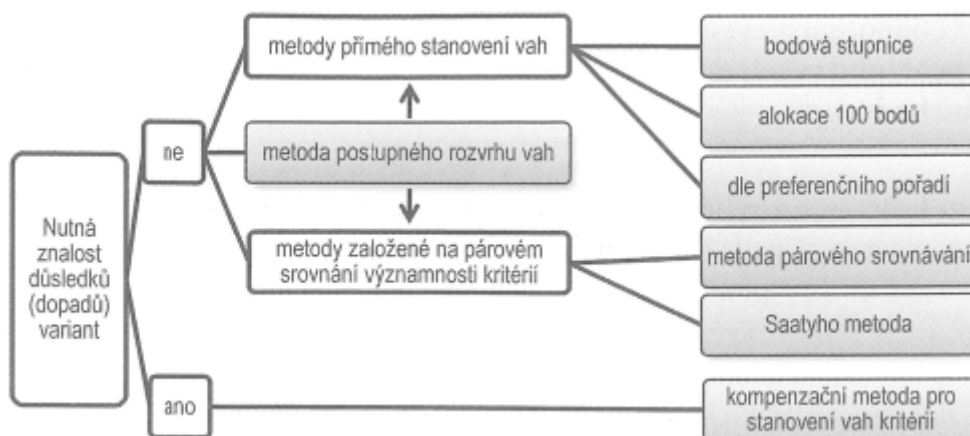
V této kapitole se nejprve seznámíme s metodami a nástroji rozhodování, které budou implementovány ve výsledném expertním systému. Poté budou popsány metody a nástroje použité pro vlastní vývoj výsledného expertního systému.

#### 3.1. *Metody a nástroje vícekritériálního rozhodování*

Jak již bylo zmíněno v předchozím textu, problém řešený v rámci této práce splňuje předpoklady pro použití metod rozhodování za jistoty. Vzhledem k povaze řešeného problému, kdy každou variantu hodnotíme podle více kritérií, se jedná o vícekritériální rozhodování za jistoty. Proto dále uvedené metody budou zaměřeny na tento typ problému.

##### 3.1.1. Metody stanovení vah kritérií

Jak uvádí Fotr [3], je u vícekritériálního rozhodování důležité nejprve stanovit váhy jednotlivých kritérií hodnocení, které jsou vyjádřeny jako číselné reprezentace jejich významnosti. Čím je kritérium významnější, tím je jeho váha vyšší a naopak. Pro dosažení srovnatelnosti vah souboru kritérií, které mohou být stanoveny různými metodami, se tyto váhy zpravidla normují tak, aby byl jejich součet roven jedné. Pro stanovení vah kritérií existuje celá řada metod. Jejich přehled je možné vidět na následujícím obrázku.



Obr. 3.1. Dělení metod pro stanovení vah kritérií

V našem případě je stanovení vah nezávislé na dopadech variant a lze tedy využít metody přímého stanovení vah nebo metody založené na párovém srovnání.

### 3.1.1.1. Metody přímého stanovení vah kritérií

Pro tyto metody platí, že při stanovování vah jednotlivých kritérií dochází k posuzování jejich významnosti přímo.

#### Bodová stupnice a alokace 100 bodů

Stanovení vah kritérií spočívá v přiřazení určitého počtu bodů ze zvolené stupnice každému kritériu, a to v souladu s tím, jak posuzovatel hodnotí význam každého kritéria. Volba stupnice závisí na diferenci významnosti jednotných kritérií a je vhodné se zamyslet před jejich stanovením nad vztahem nejvíce a nejméně významného kritéria, neboť ta budou určovat její rozpětí. Toto hodnocení bývá pro přehlednost uváděno v tabulce viz Tab. 3.1.

Kritérium	K <sub>1</sub>	K <sub>2</sub>	...	K <sub>n</sub>	Součet
Počet bodů	...	...	...	...	...
Normovaná váha	...	...	...	...	...

Tab. 3.1 Stanovení vah kritérií pomocí bodové stupnice

#### Porovnání významu kritérií pomocí jejich preferenčního pořadí

Tento způsob lze rozložit do tří kroků. Prvním krokem je stanovení preferenčního pořadí, následuje určení vah kritérií porovnáním významu kritérií s kritériem nejméně významným a nakonec je provedeno normování vah. Samotné pořadí významnosti kritérií lze stanovit přímým srovnáním (rozhodovatel určuje pořadí významnosti kritérií přímo), které se používá spíše pro menší soubor kritérií, nebo pro rozsáhlé soubory kritérií - etapovým uspořádáním (pořadí kritérií se stanovuje v několika etapách). Pro zápis se zpravidla používá stejná tabulka jako pro hodnocení pomocí bodové stupnice, ale počet bodů je dán pořadím v preferenčním uspořádání.

### 3.1.1.2. Metody stanovení vah kritérií založené na párovém srovnání

Tato metoda je založena na zjišťování preferenčních vztahů dvojic kritérií. Dvěma zástupci této metody jsou metoda párového srovnání a Saatyho metoda.



### Metoda párového srovnání

Pro každé kritérium se zjišťuje počet jeho preferencí vzhledem k ostatním kritériím souboru.

Tento postup je charakterizován následující tabulkou, která se také nazývá Fullerův trojúhelník:

Kritérium	K <sub>1</sub>	K <sub>2</sub>	K <sub>3</sub>	...	K <sub>n</sub>	Počet preferencí
K <sub>1</sub>		1	0	...	1	
K <sub>2</sub>			0	...	0	
K <sub>3</sub>					0	
...					...	
K <sub>n-1</sub>					1	
K <sub>n</sub>						

Tab. 3.2 Tabulka pro zjišťování preferencí kritérií u metody párového srovnání, viz Fotr [3]

Pro každé kritérium uvedené v řádku se zjistí, zda je preferováno před kritériem uvedeným ve sloupci (sloupcích). Pokud ano, je vidět v příslušném políčku hodnota „1“, pokud ne, hodnota „0“. Počet preferencí odpovídá počtu jedniček v řádku příslušného kritéria. Normované váhy se poté počítají podle vztahu:

$$v_i = \frac{f_i}{\sum_{i=1}^n f_i} \quad (3.1)$$

kde:

$v_i$  odpovídá normované váze i-tého kritéria;

$f_i$  je počet preferencí i-tého kritéria;

$n$  je počet kritérií.

Nevýhodou je nemožnost zahrnout do určování preferencí odlišnou míru preferencí (nelze říct, kolikrát je dané kritérium významnější než jiné kritérium).

### Saatyho metoda stanovení vah kritérií

Tato metoda odstraňuje nevýhody metody párového srovnání. Jedná se v podstatě o obdobu metody párového srovnání, ale při určování preferencí se kromě směru preference určuje také

velikost této preference. Saaty doporučuje využít pro vyjádření velikosti preferencí devítibodové stupnice opatřené deskriptory, viz následující tabulka:

Počet bodů	Deskriptor
1	Kritéria jsou stejně významná.
3	První kritérium je slabě významnější než druhé.
5	První kritérium je dosti významnější než druhé.
7	První kritérium je prokazatelně významnější než druhé.
9	První kritérium je absolutně významnější než druhé.

Tab. 3.3 Saatyho doporučená bodová stupnice s deskriptory, viz Fotr [3].

### 3.1.1.3. Metoda postupného rozvrhu vah

Když je počet kritérií větší než přibližně deset, doporučuje se použít tuto metodu, která je založena na využití stromu kritérií, tj. na seskupení kritérií v rámci celého souboru do dílčích skupin podle příbuznosti jejich věcné náplně. Pro zabezpečení srovnatelnosti je nezbytné normování vah v rámci jednotlivých skupin a jejich následné normování v rámci celého souboru kritérií.

### 3.1.1.4. Kompenzační metoda stanovení vah

Rozhodovatel (hodnotitel) si představí nejhorší a nejlepší variantu, která bude mít nejlepší, resp. nejhorší možné dopady vzhledem ke všem kritériím. Poté stanoví pořadí kritérií (určí jejich váhy) vzhledem k významnosti změny z nejméně preferované hodnoty na nejvíce preferovanou hodnotu. Výsledné váhy se normují. Tento postup se používá především v případech, kdy jsou důsledky jednotlivých variant pro dané kritérium přibližně stejné – tento aspekt poté nehraje významnou roli při rozhodování, i když rozhodovatel považuje toto kritérium za velmi důležité.

## 3.1.2. Metody hodnocení variant

Metody hodnocení variant mají obecný charakter, který není závislý na obsahové náplni jednotlivých variant. Použití jednotlivých typů se dá odvodit podle povahy kritérií. Pro soubor kritérií, který obsahuje převážně kvantitativní kritéria, je vhodné použít metody založené na párovém srovnání variant. Pro soubor kritérií spíše kvalitativní povahy je vhodné použít metody

založené na transformaci hodnot kritérií na bezrozměrnou aditivní veličinu (užitek/ohodnocení), viz Fotr [3].

### 3.1.2.1. Vícekritériální funkce utility

Většina metod ze skupiny jednoduchých metod pro stanovení hodnoty variant je založena na této metodě, resp. na jejím zjednodušení. Tato metoda bývá nazývána též jako užitková funkce nebo preferenční funkce. Principem je přiřazení utility (užitku) každé variantě vyjádřenou reálným číslem. Čím je číslo větší, tím více je daná varianta preferována. V praxi se většinou pracuje s aditivním tvarem funkce, který je vyjádřen následovně:

$$u(X) = \sum_{i=1}^n v_i \cdot u_i(x_i) \quad (3.2)$$

kde mají jednotlivé proměnné následující význam:

- $X$  varianta rozhodování;
- $u_i(x_i)$  dílčí funkce utility za jistoty i-tého kritéria;
- $x_i$  důsledek varianty vzhledem k i-tému kritériu;
- $v_i$  váha i-tého kritéria;
- $n$  počet kritérií hodnocení.

Pomocí tohoto vztahu se vyjádří dílčí utility jednotlivých variant. Celková utilita je poté určena jako vážený součet těchto dílčích hodnot. Přičemž dílčí funkce vyjadřují změnu ohodnocení v závislosti na změnách hodnoty daného kritéria. Tyto dílčí funkce mohou být rostoucí pro kritéria výnosového typu, nebo klesající pro kritéria nákladového typu. Definičním oborem dílčích funkcí užitku jsou pak intervaly hodnot jednotlivých kritérií. Je zvykem tyto funkce normovat, takže nabývají hodnot od 0 do 1. Vzhledem k normování dílčích funkcí užitku a normování vah kritérií je vícekritériální funkce také normována a nabývá hodnot z intervalu 0 (nejhorší užitek) a 1 (nejlepší užitek).

Při použití vícekritériální funkce užitku pro stanovení preferenčního uspořádání variant je nutná znalost vah kritérií. Pokud jsou použity pro konstrukci funkce užitku předem stanovené váhy, může dojít ke zkreslení výsledků (nebude respektováno rozpětí důsledků variant). V takovém případě je vhodnější použít stanovení vah kompenzační metodou.

### 3.1.2.2. Jednoduché metody stanovení užitku variant

Konstrukce dílčích funkcí užitku je poměrně náročná, a proto se v praxi často používají jednodušší metody stanovení užitku variant. Tyto metody mohou vést ke zkresleným výsledkům. Tyto metody stanovují celkové hodnocení variant jako vážený součet dílčích hodnocení variant vzhledem k jednotlivým kritériím.

$$H^j = \sum v_i \cdot h_i^j \quad (3.3)$$

pro  $j = 1, 2, \dots, m$ , kde jednotlivé proměnné představují:

- $H^j$  celkové ohodnocení j-té varianty;
- $v_i$  váha i-tého kritéria;
- $h_i^j$  dílčí hodnocení j-té varianty;
- $n$  počet kritérií hodnocení;
- $m$  počet variant.

Na základě preferenčního uspořádání je možné stanovit preferenční uspořádání s vedoucí optimální variantou.

Rozdíly mezi vícekritériální funkcí užitku a jednoduchými metodami jsou především:

- Stanovení celkového ohodnocení variant jako váženého součtu jejich dílčích ohodnocení vzhledem k jednotlivým kritériím se považuje za apriorně dané, aniž se ověřuje platnost předpokladu preferenční nezávislosti kritérií umožňujícího vyjádřit vícekritériální funkci užitku. Většina uživatelů si ani není existence tohoto předpokladu vědoma.
- Váhy kritérií se neurčují kompenzační metodou.

Jednotlivé verze metod jsou rozdílné především ve způsobu stanovení dílčích ohodnocení variant vzhledem k jednotlivým kritériím, viz dále.

#### Metoda váženého pořadí

Dílčí ohodnocení variant se stanovuje vzhledem k jednotlivým kritériím podle pořadí variant vzhledem k těmto kritériím. Používá se následující vzorec:

$$h_i^j = m + 1 - p_i^j \quad (3.4)$$

kde:

$p_i^j$  je pořadí j-té varianty vzhledem k j-tému kritériu;

$h_i^j$  dílčí hodnocení j-té varianty;

$m$  je počet variant.

Nedostatkem této metody je její značná hrubost, která plyne z daného vztahu. Použitelná je zejména v případech, kdy soubor kritérií je převážně kvalitativní povahy.

### **Metoda expertního stanovení dílčího ohodnocení**

U této metody je dílčí hodnocení stanovováno přímo rozhodovatelem, který vystupuje v roli experta. Hodnocení je dáno jako přiřazení bodů jednotlivým variantám ze zvolené bodové stupnice vzhledem k jednotlivým kritériím. Výhodou této metody je kromě jednoduchosti a srozumitelnosti pro rozhodovatele také respektování nelinearity závislosti dílčích ohodnocení variant na jejich důsledcích, pokud je zvolena dostatečně jemná bodová stupnice. Nevýhodou je vyšší náročnost na hodnotitele.

### **Metoda lineárních dílčích funkcí užitku**

Dílčí ohodnocení se stanovuje podle typu kritéria. Pro kvalitativní kritéria je postup stejný jako u metody expertního stanovení dílčích ohodnocení, tj. přiřazením bodů ze zvolené bodové stupnice. U kvantitativních kritérií se nejhorší hodnotě každého kritéria přiřadí hodnocení 0 ( $x_i^0$ ) a nejlepší 1 ( $x_i^*$ ) a spojnice těchto bodů pak zobrazuje lineární dílčí funkci užitku. Předpokladem tohoto postupu je lineární tvar dílčích funkcí užitku. Vztah pro hodnocení u kvantitativních kritérií je dán jako:

$$h_i^j = \frac{x_i^j - x_i^0}{x_i^* - x_i^0} \quad (3.5)$$

Výhodou této metody je snížení subjektivity stanovení dílčích ohodnocení variant pro kvantitativní kritéria. Dle Fotra [3], „předpoklad linearit dílčích funkcí je zpravidla přijatelný, neboť některé empirické výzkumy ukazují, že takto získané celkové ohodnocení je zpravidla

dobrou aproximací ohodnocení plynoucího z funkce užitku respektující možné nelinearity dílčích funkcí užitku“.

### **Metoda bazické varianty**

Dílčí ohodnocení variant vzhledem jednotlivým kritériím se stanovuje porovnáním hodnot důsledků variant s hodnotami tzv. bazické varianty. Bazická varianta představuje variantu, která dosahuje nejlepších hodnot kritérií z daného souboru variant, nebo varianta, která pro jednotlivá kritéria nabývá cílových hodnot.

Pro kritéria výnosového typu se dílčí ohodnocení variant stanoví jako:

$$h_i^j = \frac{x_i^j}{x_i^b} \quad (3.6)$$

kde:

$x_i^b$  je důsledek bazické varianty pro i-té kritérium;

$h_i^j$  dílčí hodnocení j-té varianty.

Pro kritéria nákladového typu je výpočet dílčího ohodnocení variant dán jako:

$$h_i^j = \frac{x_i^b}{x_i^j} \quad (3.7)$$

Tato metoda je využitelná především pro soubor kvantitativních kritérií.

### **3.1.2.3. Metody založené na párovém srovnání variant**

Tyto metody jsou vhodné především pro soubor kvalitativních kritérií nebo soubor smíšených kritérií, kde kvantitativní kritéria převažují. Základem pro stanovení preferenčního uspořádání variant jsou výsledky párového srovnání těchto variant vzhledem k jednotlivým kritériím.

### **Saatyho metoda**

Celkové ohodnocení variant je dáno jako vážený součet dílčích ohodnocení variant vzhledem k jednotlivým kritériím, viz vzorec 3.3.

Stanovení dílčích ohodnocení jednotlivých variant je analogické jako určení vah kritérií Saatyho metodou. Jediným rozdílem je, že srovnávanými objekty jsou varianty a ne kritéria. Výhodou

této metody je její použitelnost především pro smíšené soubory kritérií (obsahující kvalitativní i kvantitativní kritéria) a relativní jednoduchost a srozumitelnost pro uživatele.

### ***3.2. Metody a technologie použité pro tvorbu expertního systému***

V této kapitole jsou v první části uvedeny obecné metodiky vývoje nejen informačních systémů. Specifické aspekty vývoje expertních systémů jsou uvedeny ve druhé části kapitoly. Protože tyto aspekty vyplývají z architektury expertních systémů, je ve druhé části kapitoly nejprve uvedeno krátké obecné seznámení s expertními systémy, tj. z čeho se skládají, jaká jsou jejich specifika oproti klasickým informačním systémům atd.

#### **3.2.1. Metodiky vývoje (expertních) systémů**

Pojmem metodika vývoje softwaru se obecně označuje soubor znalostí a postupů, který je používán během celého vývoje informačního systému. Existuje celá škála takových metodik a každá má své výhody a nevýhody pro vývoj daného typu informačního systému. Obecně neexistuje jejich ucelený popis a srovnání nebo doporučení, kterou metodiku lze kdy použít. Navíc se některé metodiky zaměřují pouze na specifické fáze vývoje. Není tedy vždy snadné vybrat pro daný typ systému tu nejvhodnější.

Jako hlavní kritérium pro výběr metodiky se používá její „podrobnost“. Podle tohoto kritéria lze metodiky dělit na:

- Agilní – jedná se o minimálně dostatečné metodiky pro zvládnutí vývoje daného systému. Představují většinou pouze obecné rady, postupy a doporučení, kladou důraz hlavně na produkt a méně již na postupy, které vedou k jeho tvorbě.
- Tradiční - jedná se o obsáhlé a podrobné metodiky. Představují většinou podrobný popis a vývoj několika různých modelů specifikujících daný systém.

Při použití tradičních metodik je fixní cílová funkcionalita, které má být dosaženo použitím proměnných zdrojů v proměnném čase. Při použití agilních metodik je dán čas a zdroje a cílová funkcionalita systému se během vývoje může měnit.

Pro tuto práci, kdy jsou dány požadavky na výsledný systém, budou využity některé z tradičních metodik vývoje informačních systémů, viz následující kapitola.

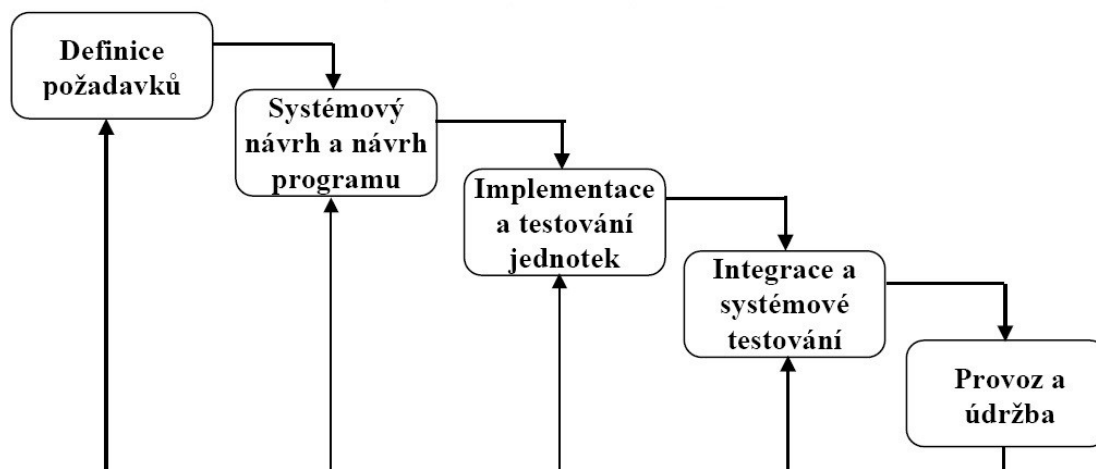
### **3.2.1.1. Tradiční metodiky vývoje software**

Ve většině případů jsou modely potřebné v průběhu vývoje předem dané. Obecně jsou dány požadavky na opakovatelnost procesů a znovupoužitelnost modelů, možnost definovat všechny požadavky předem, neměnitelnost definovaných požadavků. Další konkrétní požadavky jsou pak dány typem metodiky. Jako hlavní zástupci tradičních metodik budou dále uvedeny metodiky vodopád, inkrementální vývoj, prototypování a spirálový vývoj.

#### **Vodopád**

Metodika vývoje vodopád je inspirována analogií z reálného světa – vodopádem – a je jednou z nejznámějších metodik vývoje informačních systémů založených na modelu vodopád. Projekt lineárně postupuje („přelévá se“) mezi předem danými fázemi. Jednotlivé části mají specifikovány své vstupy a výstupy. Přechody mezi fázemi jsou přirozené milníky vývoje a tak lze snadno sledovat dosažený pokrok ve vývoji. Přechod z jedné části do druhé je možný pouze tehdy, jestliže je daná část hotová. Tento model je využitelný zejména u jednorázových, krátkodobých a dobře specifikovaných projektů, u kterých nedochází ke změnám v požadavcích během vývoje. Model vodopádu je značně rozšířeným způsobem vývoje aplikací a existuje celá řada metodik na něm založených. Jako příklad lze uvést inkrementální model, který je popsán dále. Obecný model vodopádu je ilustrován na následujícím obrázku, viz Obr. 3.2.





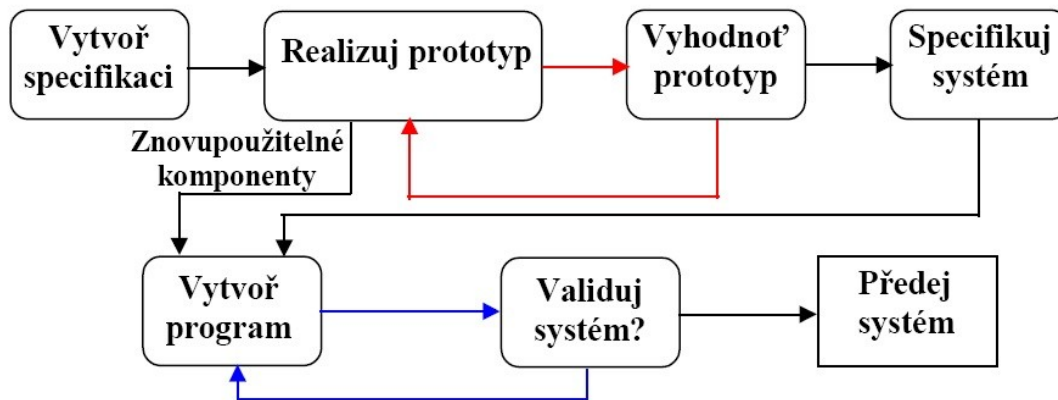
Obr. 3.2. Model vodopád, viz Zendulka [8]

### Inkrementální vývoj

Metodika inkrementálního modelu je modifikací modelu vodopád, ale eliminuje jednosměrnost tohoto modelu. Inkrementální model je založen na rozdělení celého vývoje systému do menších samostatných částí – iterací. Každá iterace je řešena jako menší samostatný projekt založený na modelu vodopád. Výhodou tohoto přístupu oproti obecné metodice vodopádu je rychlejší dostupnost první funkční verze systému. V dalších iteracích je tato první verze dále rozšiřována až do cílové podoby systému, navíc po dokončení každé iterace existuje funkční aplikace.

### Prototypování

V rámci této metodiky je co nejrychleji vytvořena první verze systému (prototyp), která odpovídá hlavním požadavkům na cílový systém. Poté je konzultován s uživateli a na základě této konzultace se zpřesní požadavky na systém. Během fáze hodnocení prototypu je nutná komunikace s uživateli. Jednotlivé fáze vývoje pomocí prototypování je možné vidět na následujícím obrázku, viz Obr. 3.3.



Obr. 3.3. Model prototypování, viz Zendulka [8]

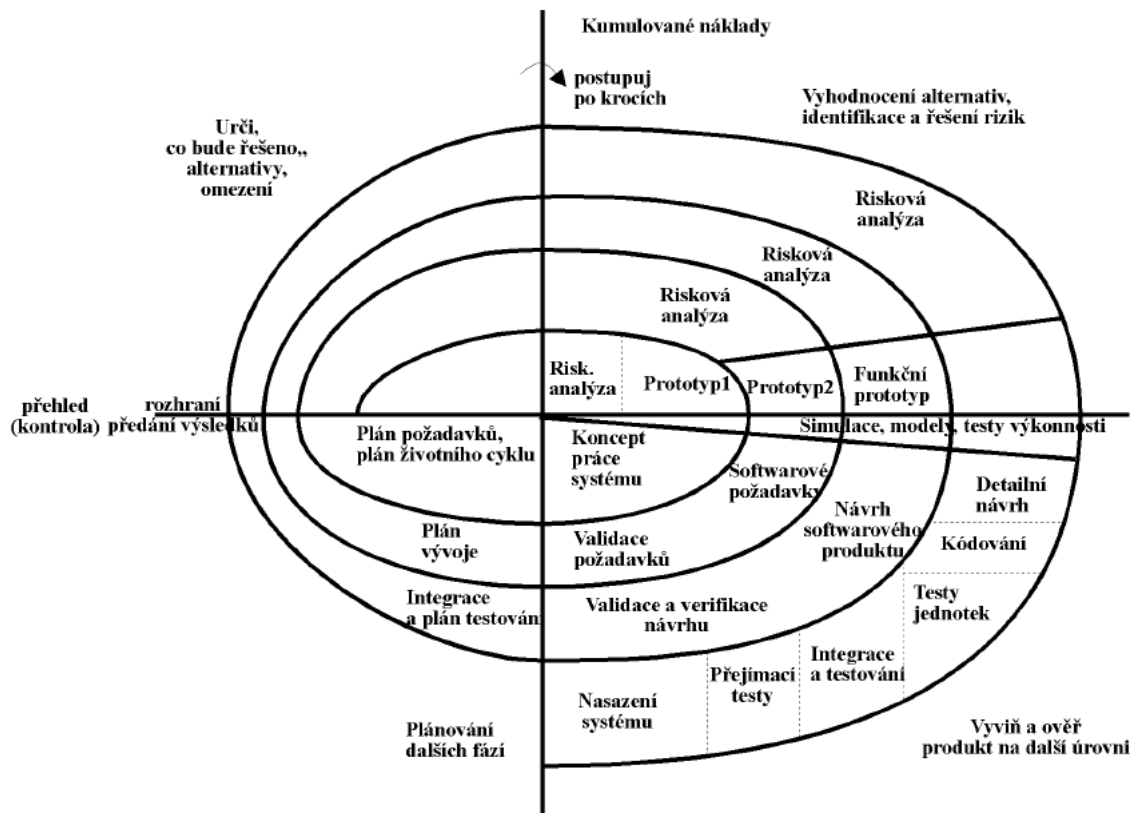
### Spirálový vývoj

Jedná se o metodiku, kde vývoj systému probíhá podle spirálového modelu. Metodika kombinuje vlastnosti vodopádového vývoje i prototypování. Poprvé byl tento model popsán v roce 1988 v článku „A Spiral Model of Software Development and Enhancement” Barryho Boehma. Metodika vývoje podle spirálového modelu je určena zejména pro velké a nákladné projekty. Jedná se o iterativní vývoj, tj. celý projekt je rozložen do několika po sobě jdoucích iterací. Každá iterace se řeší jako samostatný projekt metodikou vodopádu a začíná definicí požadavků na tuto iteraci a končí po úspěšném testování předáním iterace.

Spirálový model je používán zejména v případě, že na začátku vývoje systému nemáme veškeré potřebné informace. Potřebné informace získáváme v průběhu jednotlivých iterací (z konzultací prototypů se zákazníkem). Díky těmto informacím se zpřesňují požadavky a celý vývoj se přibližuje ke konečné podobě systému. Každá iterace obnáší šest kroků:

- určení cílů, alternativ a omezení;
- identifikace a řešení rizik;
- vyhodnocení alternativ;
- vývoj a testování aktuální iterace;
- plánování další iterace;
- rozhodování o postupu do další iterace.

Schéma spirálového modelu je vidět na následujícím obrázku, viz Obr. 3.4.



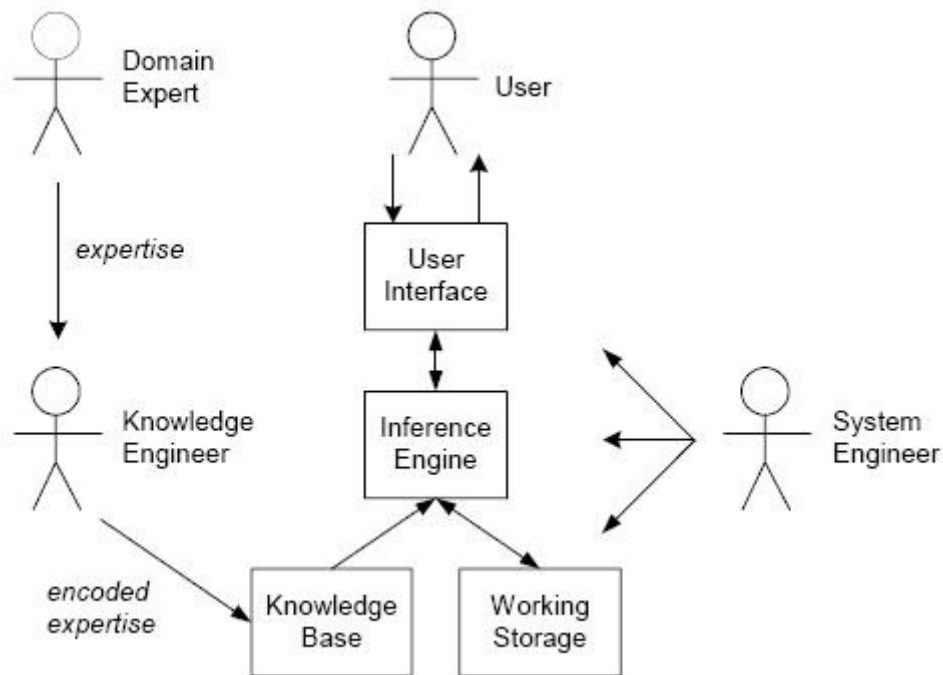
Obr. 3.4. Spirálový model, viz Sochor [9]

### 3.2.2. Expertní systémy

Expertní systémy se objevily v 70. letech 20. století a našly své uplatnění v mnoha oblastech od kontrolních systémů až po pomocné systémy. Jsou využívány zejména pro účely diagnostik a předpovědí. Typický způsob použití je založen na komunikaci s uživatelem za účelem získání faktů, které jsou následně použity pro generování předpovědi nebo diagnózy.

V literatuře se často setkáváme také s pojmem znalostní systém, který je uváděn jako obecnější pojem než expertní systém. Pojem expertní systém může být chápán jako speciální typ znalostního systému, který se vyznačuje některými specifickými rysy jako používáním expertních znalostí, používáním vysvětlovacího mechanismu atd. Často jsou však tyto pojmy chápány jako synonyma.

### 3.2.2.1. Architektura expertních systémů



Obr. 3.2 Komponenty expertních systémů a uživatelské rozhraní, viz Meritt [5].

Jak je vidět z obrázku 3.2, hlavními komponentami expertních systémů jsou:

- báze znalostí (Knowledge base) – deklarativní reprezentace odborné znalosti, často reprezentována pomocí IF THEN pravidel;
- pracovní sklad (Working storage) – data specifická pro problém, který má být řešen;
- inferenční (odvozovací) mechanismus – kód v jádru systému, který odvozuje doporučení na základě báze znalostí a dat, která jsou pro daný problém specifická a jsou uložena v pracovním skladu;
- uživatelské rozhraní – kód, který zabezpečuje komunikaci mezi uživatelem a systémem.

Pro pochopení architektury a návrhu expertních systémů je také důležité rozumět klíčovým rolím osob, které se nějakým způsobem podílejí na tvorbě a údržbě systému nebo s ním komunikují. Jsou to následující role:

- doménový expert – osoba, která vystupuje v roli experta pro oblast, ve které má být daný problém řešen;

- znalostní inženýr – osoba, která převede expertní znalosti na deklarativní formu, která může být využita v rámci expertního systému;
- uživatel – osoba, která bude komunikovat se systémem za účelem konzultace, která by mohla být poskytnuta doménovým expertem.

### **3.2.2.2. Implementace expertních systémů**

Problematikou implementace expertních systémů se zabývá znalostní inženýrství. Znalostní inženýrství má mnoho společných aspektů jako softwarové inženýrství. Celá implementace expertního systému se dá popsat následujícími kroky

- výběr hardwaru a softwaru;
- návrh uživatelského rozhraní;
- získání a reprezentace znalostí;
- implementace;
- validace a verifikace.

### **3.2.2.3. Životní cyklus expertních systémů**

Model životního cyklu expertního systému je většinou založen na kombinaci inkrementálního vývoje a rychlého prototypování (viz kapitola 3.2.1.1) a obsahuje následující etapy:

1. Analýza problému – cílem je posuzování vhodnosti aplikace expertního systému při řešení daného problému a dostupnosti zdrojů (lidských expertů nebo alternativních zdrojů znalostí).
2. Specifikace požadavků – obsahuje zpravidla části jako charakteristika problému, profil uživatelů, cíle projektu, vstupy a výstupy systému, pomocné funkce, hardwarová omezení, externí rozhraní implementační priority, metody validace a verifikace, požadavky na dokumentaci.
3. Předběžný návrh – měl by obsahovat výběr typu reprezentace znalostí (pravidla, logika, struktury), výběr metod usuzování (měl by vycházet z volby reprezentace znalostí), výběr nástrojů atd.
4. Rychlé prototypování a vyhodnocování – v praxi je často založeno na využití například Prologu. Jeho cílem je rychlé vytvoření fungujícího prototypu systému. Poté je prototyp vyhodnocen a podle výsledků jsou provedeny potřebné úpravy. Už první verze prototypu

by měla mít funkční uživatelské rozhraní a základní podmnožinu znalostí, aby bylo možné ji zhodnotit. Stejně jako u prototypování v případě klasických informačních systémů doporučuje opuštění (zahození) prototypové verze a započetí implementace expertního systému na základě výsledků hodnocení prototypu od počátku.

5. Konečný návrh – vzniká na základě výsledků hodnocení prototypu
6. Implementace – implementace finálního systému.
7. Validace a verifikace – testování výsledné aplikace.
8. Změny návrhu.
9. Údržba.

V rámci inkrementálního vývoje se etapy 6, 7 a 8 iteračně opakují.

#### **3.2.2.4. Výhody a nevýhody expertních systémů**

Expertní systém je speciálním případem informačního systému, který má zastupovat živého člověka v roli experta. Proto budou výhody a nevýhody zaměřeny na srovnání expertního systému jak s běžným informačním systémem, tak s člověkem v roli experta.

##### **Výhody**

- Velkou výhodou expertních systémů oproti běžným systémům je, že dokáží výsledek jednoznačně zdůvodnit.
- Narozdíl od experta – člověka, poskytuje stále stejné výsledky, může pracovat neustále, rozhodování není ovlivněno únavou, stresem, ani jinými faktory.

##### **Nevýhody**

- Při vzniku expertního systému může docházet ke zkreslení dat, protože člověk v roli experta nedokáže své vědomosti předat přímo, ale je odkázaný na prostředníka – expertního inženýra, a navíc expert většinou nedokáže dostatečně podrobně popsat všechny faktory, které mají vliv na jeho rozhodnutí.
- Na lidské rozhodování mají vliv i znalosti a schopnosti, které s problémem na první pohled přímo nesouvisí a které se nedají implementovat do expertního systému. Příkladem je všeobecný přehled, životní zkušenost a moudrost atd.

## **4. Analýza řešené problematiky a návrh expertního systému**

### **4.1. Analýza současného stavu**

V současné době jsou data o volných pracovních místech jednotlivých úřadů práce dostupná na stránkách Ministerstva práce a sociálních věcí, respektive na Integrovaném portále Ministerstva práce a sociálních věcí („<http://portal.mpsv.cz/sz>“). Veškerá vstupní data pro tuto práci jsou získána z tohoto portálu v době 15.3.2011.

#### **4.1.1. Úřady práce – současná praxe**

##### **4.1.1.1. Evidence a vyhledávání**

Každá nabídka práce evidovaná na kterémkoliv úřadě práce v ČR je také evidována v databázi Integrovaného portálu. Zaměstnavatelé mohou své nabídky práce vkládat přímo přes rozhraní portálu nebo je mohou evidovat u příslušného pracovního úřadu. Jednotlivé nabídky práce je možné na portále vyhledávat několika způsoby.

#### **Možnosti vyhledávání práce**

Pro lidi, kteří nejsou příliš zdatní v práci s výpočetní technikou, je určen základní formulář pro vyhledávání. Zde uživatel může zadat základní parametry pro vyhledávání a to:

- okres – NUTS 3 (Nomenklatura územních statistických jednotek), v rámci které mají být nabídky vyhledávány;
- vzdělání – nejvyšší dosažené vzdělání.

Rozšířené vyhledávání pracovních příležitostí je určeno spíše pokročilejším uživatelům výpočetní techniky. Je zde možné kombinovat různé způsoby vyhledávání.

Kromě informací pro občany a zaměstnavatele je zde možné nalézt v sekci „Zahraniční zaměstnanost“ informace o možnosti zaměstnání v zahraničí, případně o možnosti zaměstnání cizích státních příslušníků v ČR. V sekci statistiky je možné nalézt informace o počtu pracovních míst v jednotlivých okresech, statistiky zaměstnanosti atd.

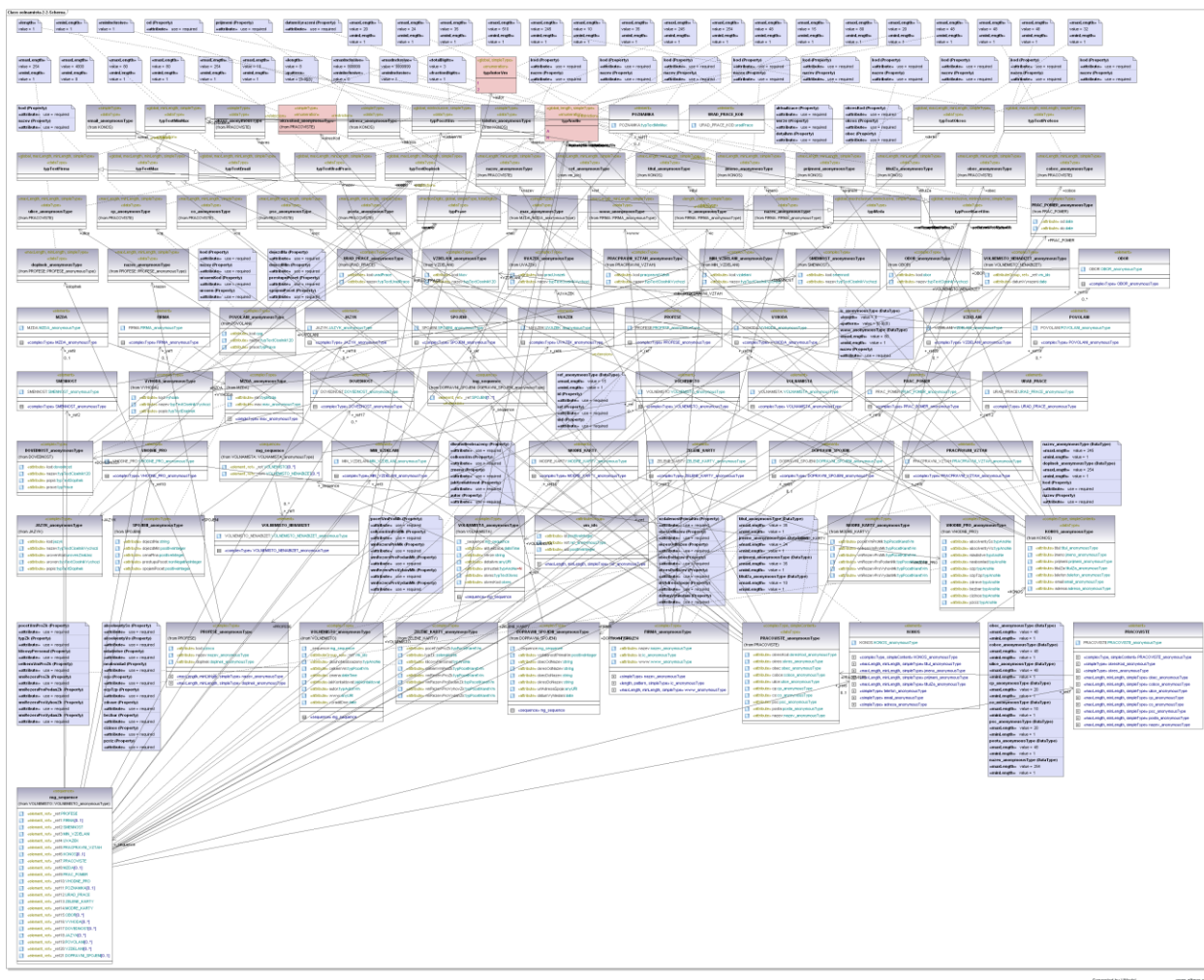
#### 4.1.1.2. Dostupnost a reprezentace vstupních dat

V sekci „Ke stažení“ je možné stáhnout veškeré informace a data a prohlížet je bez nutnosti online připojení. Pro tuto práci jsou relevantní data dostupná v sekci „Volná místa evidovaná Úřadem práce ČR“ pod názvem „Volná místa za celou ČR“ a „Formát XML“ (formát XML je popsán dále v této kapitole). Zde je možné stáhnout zip soubor, který obsahuje mimo jiné soubor ve formátu XML obsahující veškeré nabídky práce evidované na úřadech práce k určitému datu. Při poslední změně ve formátu XML souboru byla provedena náhrada číselníku KZAM za číselník CZ-ISCO a jednotlivé XML soubory teď odpovídají XSD šablonám (formát XSD je popsán dále v této kapitole) verze 2.2 dostupných na adresách:

- „<http://portal.mpsv.cz/xml/exportvm/volnamista-2-2.xsd>“ obsahující strukturu souboru s nabídkami práce.
- „<http://portal.mpsv.cz/xml/ciselniky-1-1.xsd>“ obsahující strukturu potřebných číselníků.

Datová struktura šablony pro soubor nabídek práce je ilustrována následujícím obrázkem vygenerovaným nástrojem UModel společnosti Altova (obrázek je ve své původní velikosti součástí příloh na CD, viz Seznam příloh na CD):





Obr 4. 1Datový model XSD souboru nabídek práce

## XML

Extensible Markup Language (zkráceně XML, česky rozšiřitelný značkovací jazyk) je obecný značkovací jazyk, který „byl vyvinut a standardizován konsorciem W3C. Je zjednodušenou podobou staršího jazyka SGML. Umožňuje snadné vytváření konkrétních značkovacích jazyků (tzv. aplikací) pro různé účely a různé typy dat. Zpracování XML je podporováno řadou nástrojů a programovacích jazyků. Jazyk je určen především pro výměnu dat mezi aplikacemi a pro publikování dokumentů, u kterých popisuje strukturu z hlediska věcného obsahu jednotlivých částí, nezabývá se vzhledem. Prezence dokumentu (vzhled) může být definována pomocí kaskádových stylů. Další možností zpracování je transformace do jiného typu dokumentu, nebo do jiné aplikace XML.“, viz [10].

## **XSD**

XML Schema Definition (XSD) je „XML schéma, které popisuje strukturu XML dokumentu. Je alternativa k jinému popisu struktury XML - DTD. Definuje místa v dokumentu, na kterých se mohou vyskytovat různé elementy, definuje atributy, definuje, které elementy jsou potomky jiných elementů; definuje pořadí elementů, počty elementů; definuje, zda element může být prázdný, nebo zda musí obsahovat text; definuje datové typy elementů a jejich atributů; definuje standardní hodnoty elementů a atributů.“, viz [11].

### **Současný stav datových struktur**

Zde je popsán stav datových struktur poskytovaný Integrovaným portálem v době vzniku této práce.

S ohledem na sdělení Českého statistického úřadu ze dne 16. června 2010 o zavedení Klasifikace zaměstnání (CZ-ISCO) byl 17. 1. 2011 nahrazen číselník KZAM číselníkem CZ-ISCO. Podrobné informace o klasifikaci jsou umístěny na stránkách Českého statistického úřadu a stránkách Metodického centra pro klasifikaci zaměstnání CZ-ISCO.

V souvislosti s náhradou číselníku KZAM za číselník CZ-ISCO byl 17. 1. 2011 změněn formát souborů XML s volnými místy ke stažení na verzi 2.2 (viz soubor s popisy položek - <http://portal.mpsv.cz/xml/exportvm/volnamista-2-2.xsd> a soubor s hodnotami všech číselníků - <http://portal.mpsv.cz/xml/ciselniky-1-1.xsd>), která je položkově zpětně kompatibilní s předchozí verzí 2.0.

Formát 2.2 obsahuje následující změny:

- přechod na kódování profese pomocí CZ-ISCO namísto KZAM,
- rozšíření položky doplněk na 254 znaků (položka doplněk, je-li zadána, přebírá roli hlavního názvu profese),
- doplnění informací o nabídce pro modré karty.

V souvislosti se zavedením nové klasifikace zaměstnání bylo upraveno formátování výpisu nabídky volného místa. Pokud má volné místo definovaný doplněk profese, je uváděn před

výpisem klasifikace podle CZ-ISCO. Položka doplněk má v takové situaci význam hlavního názvu profese a tedy hlavního nadpisu volného místa.

22. 2. 2011 bylo opraveno formátování desetinných čísel v položkách praxe u dovedností nebo povolání. V případě výskytu desetinných čísel se jako oddělovač objevovala čárka namísto tečky.

### **Aktualizovaný stav datových struktur**

S ohledem na účinnost zákona o Úřadu práce České republiky a o změně souvisejících zákonů (zákon č. 73/2011 Sb.) byl 1. 4. 2011 změněn formát souboru s pracovišti Úřadu práce ke stažení.

Nový formát reflektuje organizační strukturu Úřadu práce ČR (generální ředitelství, krajské pobočky, kontaktní pracoviště). V souvislosti se změnou formátu byla do souboru doplněna i pracoviště státní sociální podpory. Další informace o Úřadu práce ČR jsou umístěny na stránce Úřadu práce České republiky.

Tyto změny nebyly v době vzniku práce známy a nejsou zohledněny ve výsledném systému. Nicméně na funkčnost systému by neměly mít vliv.

### **4.1.2. Hodnocení současné situace**

V současné době jsou nabídky práce evidovány jak na jednotlivých pracovních úřadech, tak i v centrální databázi Ministerstva práce a sociálních věcí. Tato data jsou dostupná v rámci Integrovaného portálu Ministerstva práce a sociálních věcí. Vyhledávání s použitím příslušného rozhraní tohoto portálu umožňuje uchazečům základní a pokročilé vyhledávání pracovních nabídek. Neumožňuje však při žádném typu vyhledávání zohlednit preference uživatelů (tj. zohlednit preference jednotlivých vyhledávacích kritérií), čímž by se zpřesnil výsledek vyhledávání a užitek vyhledávání pro uživatele by se zvýšil.

#### **4.1.2.1. SWOT analýza současného stavu**

SWOT analýza je nástroj strategického plánování. Používá se při hodnocení silných a slabých stránek, příležitostí a hrozeb, které souvisejí s daným projektem. Základem je klasifikace

jednotlivých faktorů, které jsou rozděleny do čtyř výše uvedených základních skupin. Celá analýza spočívá v hodnocení současného stavu – vnitřního prostředí a současné situace okolí – vnějšího prostředí. Ve vnitřním prostředí se hledají silné a slabé stránky. Ve vnějším prostředí se hledají příležitosti a hrozby.

Výsledky analýzy bývají využity pro stanovení strategie a budoucího rozvoje. Stanovení strategie spočívá v hledání vzájemné synergie mezi silnými a slabými stránkami, příležitostmi a silnými stránkami apod. Výsledná strategie poté spadá do jedné z následujících skupin:

- MAX-MAX strategie – maximalizace silných stránek – maximalizace příležitostí;
- MIN-MAX strategie – minimalizace slabých stránek – maximalizace příležitostí;
- MAX-MIN strategie – maximalizace silných stránek – minimalizace hrozeb;
- MIN-MIN strategie – minimalizace slabých stránek – minimalizace hrozeb.

V následující tabulce, viz Tab. 4.1 je uvedena SWOT analýza současného stavu vyhledávacího systému pracovních nabídek v rámci Integrovaného portálu Ministerstva práce a sociálních věcí.

<b>Silné stránky</b>	<b>Slabé stránky</b>
<ul style="list-style-type: none"> <li>- dlouhodobě existující a fungující systém =&gt; velké zkušenosti</li> <li>- existující uživatelská skupina</li> <li>- možnosti stahování strukturovaných podkladových dat ve formě XML (XSD)</li> <li>- existující a funkční základní vyhledávací služby</li> </ul>	<ul style="list-style-type: none"> <li>- velmi robustní systém =&gt; pomalé reakce na případné nutné změny</li> <li>- nutnost častých úprav plynoucích z legislativy</li> </ul>
<b>Příležitosti</b>	<b>Hrozby</b>
<ul style="list-style-type: none"> <li>- poskytovat některé služby přes rozhraní webových služeb</li> <li>- zohlednit preference uživatelů pro jednotlivá vyhledávací kritéria</li> </ul>	<ul style="list-style-type: none"> <li>- nedostatečná zpětná odezva uživatelů může vést k jejich odchodu</li> <li>- nutnost zpětné kompatibility mezi změnami zpožďuje a prodražuje jejich implementaci</li> </ul>

**Tab. 4. 1. SWOT analýza současného stavu**

#### **4.1.2.2. Příležitosti využitelné v navrhovaném systému**

Jak již bylo zmíněno, cílem systému vyvíjeného v rámci této práce je zohlednit při vyhledávání pracovních nabídek preference uživatelů. Tento cíl byl také identifikován jako jedna z příležitostí

dalšího vývoje již existujícího systému. Vývoj takového systému je umožněn díky jedné ze silných stránek systému a to možnosti stahování strukturovaných podkladových dat ve formě XML (XSD). Vyvíjená aplikace by v budoucnu mohla být dostupná online přes rozhraní běžného webového prohlížeče, což by kladlo důraz na přenositelnost celého systému.

## **4.2. Návrh rozhodovacího procesu**

Teorie rozhodovacího procesu byla již popsána v předchozím textu (viz kapitola 2.5) a jednotlivé použitelné metody a nástroje byly popsány v kapitole 3.1. Tato kapitola se bude zabývat postupně volbou vhodných kritérií pro vyhledávání pracovních nabídek a výběrem metod pro stanovení vah kritérií a hodnocení variant.

### **4.2.1. Výběr kritérií pro vyhledávání**

Maximální množina použitelných kritérií je dána dostupnými daty, respektive XSD šablonou. Jak již bylo dříve zmíněno, základní funkčnost vyhledávání Integrovaného portálu Ministerstva práce a sociálních věcí nabízí pouze omezené možnosti a rozšířené vyhledávání je svou složitostí určeno spíše pro pokročilejší uživatele výpočetní techniky. Proto v rámci této práce bude vytvořen systém jako kompromis mezi základním a rozšířeným vyhledáváním. Jednotlivá kritéria jsou vybrána z dostupných dat.

Jako kritéria vhodná pro posouzení jednotlivých pracovních nabídek byla vybrána:

- Obor – obor volného místa;
- Okres – okres (NUTS 3) pro vyhledávání voleného místa (pokud je soubor exportem volných míst pouze pro jeden okres, je zde uveden kód okresu);
- Mzda – minimální požadovaná mzda v Kč za měsíc;
- Vzdělání – požadované vzdělání.

Jako další alternativy kritérií dostupné v datech by přicházely v úvahu například:

- Dovednost – požadované dovednosti;
- Úvazek – pracovní úvazek;
- Jazyk - požadovaný jazyk;
- Směnnost – typ směn;
- Úřad práce – úřad, u kterého mají být nabídky vyhledány;
- Výhoda - nabízené výhody.

#### 4.2.2. Výběr metody stanovení vah kritérií

V tomto případě není stanovení vah závislé na znalosti dopadů variant, proto lze využít pro stanovení vah kritérií metodu přímého stanovení vah kritérií nebo metody založené na párovém srovnání významnosti kritérií. Metody párového srovnání jsou co do práce uživatele náročnější (uživatel musí porovnat každý pár kritérií), což by mohlo znesnadnit používání systému, proto bude v našem případě použita metoda přímého stanovení vah kritérií. Při větším počtu kritérií by bylo vhodné využít kombinace metody přímého stanovení vah kritérií a metody postupného rozvrhu vah, což však vzhledem k počtu kritérií (méně než 10) nebude v práci realizováno.

#### 4.2.3. Výběr metody hodnocení variant

Metody párového srovnání by byly zbytečnou zátěží uživatele. Proto bude vybrána některá z jednoduchých metod hodnocení variant, viz Tab. 4.1.

Metoda	Vhodnost	Omezení (nevýhoda)
<b>váženého pořadí</b>	pro kvalitativní kritéria	neodráží rozdíly mezi hodnotami u kvantitativních kritérií
<b>přímého stanovení dílčích ohodnocení</b>	pro kvalitativní i kvantitativní kritéria	vyšší subjektivita a náročnost na hodnotitele
<b>lineární dílčí funkce užitku</b>	pro kvalitativní i kvantitativní kritéria	předpokládá linearitu dílčích funkcí užitku
<b>bazické varianty</b>	pro kvantitativní kritéria	předpokládá linearitu pro výnosová kritéria a nelineární průběh pro nákladová kritéria

Tab. 4.1. Srovnání metod stanovení hodnoty variant, viz Fotr [3].

V souboru vybraných kritérií se vyskytují jak kvantitativní, tak kvalitativní kritéria. Přímé stanovení dílčích ohodnocení by bylo vzhledem k velkému počtu variant nereálné, a proto bude pro hodnocení variant použita metoda lineárních dílčích funkcí užitku.

#### 4.2.4. Výběr vývojového cyklu

Jak bylo uvedeno v kapitole zabývající se životním cyklem expertních systémů, v praxi se při jejich vývoji většinou používá kombinace inkrementálního vývoje a rychlého prototypování. Stejný postup bude uplatněn i pro tuto práci.

### ***4.3. Návrh expertního systému***

Na základě výsledku analýzy současného stavu, především pak výsledků SWOT analýzy, byly specifikovány následující požadavky na vyvíjený systém:

- Možnost nahrát do systému svá data ve formátu XML v odpovídající struktuře.
- Kromě možnosti zadávání hodnot kritérií umožní systém zadat také váhy jednotlivých kritérií odpovídající preferencím uživatelů.
- Zobrazení výsledků vyhledávání v přijatelné formě s možností podrobného výpisu.
- Systém bude přenositelný, aby umožňoval budoucí použití přes rozhraní webového prohlížeče na různých systémových platformách (Windows, Mac OS, Linux).

#### **4.3.1. Výběr hardwaru a softwaru**

Tato kapitola se bude věnovat výběru hardwaru a softwaru pro implementaci výsledné aplikace.

##### **4.3.1.1. Aplikační logika**

Celý systém by měl být zejména kvůli požadavku na přenositelnost vyvíjen v programovacím jazyce Java, viz kapitola 4.3.1.1.1. Požadavek na možnost online přístupu by v budoucnu mohl být naplněn například použitím technologie Java Web Start, což je technologie, která slouží k jednoduché distribuci Java aplikací na internetu. Jedná se o distribuci a spouštění aplikací pomocí kliknutí na odkaz přímo z webových stránek. Další výhodou je, že uživatel má vždy k dispozici poslední verzi aplikace, čímž se vyhne problémům s aktualizacemi programu.

Část aplikační logiky by měla být naprogramována s využitím deklarativního programovacího jazyka prolog, viz kapitola 4.3.1.1.2. Aby byl dodržen požadavek přenositelnosti, je nutné najít vhodný interpret tohoto jazyka, který je kompatibilní s Javou, respektive interpret prologu pro Javu. V rámci této práce byly zkoumány a vyzkoušeny SWI-Prolog a JIProlog. SWI-Prolog nabízí větší možnosti pro vývoj, ale pro komunikaci s Javou jsou využity JNI (Java Native Interface) knihovny a při pokusech jeho použití na systému Mac OS jsem neuspěl (nebyl dostupný návod, ani alternativa pro JNI knihovny). Navíc instalace výsledného systému by vyžadovala i distribuci příslušných knihoven. Oproti tomu JIProlog je sice distribuován pod Shareware licencí, ale jedná se o multiplatformní interpret prologu pro Javu, který není závislý

na nativním kódu. Nebyl problém s jeho použitím jak na Windows systému, tak na Mac OS systému.

Na základě popsaných výhod/nevýhod obou interpretů jazyka prolog byl pro vlastní implementaci vybrán JIProlog.

#### **4.3.1.1.1. Programovací jazyk Java**

Java je „objektově orientovaný programovací jazyk, který vyvinula firma Sun Microsystems a představila 23. května 1995. Java je jedním z nejpoužívanějších programovacích jazyků na světě. Díky své přenositelnosti je používán pro programy, které mají pracovat na různých systémech počínaje čipovými kartami (platforma JavaCard), přes mobilní telefony a různá zabudovaná zařízení (platforma Java ME), aplikace pro desktopové počítače (platforma Java SE) až po rozsáhlé distribuované systémy pracující na řadě spolupracujících počítačů rozprostřené po celém světě (platforma Java EE). Tyto technologie se jako celek nazývají platforma Java. Dne 8. května 2007 Sun uvolnil zdrojové kódy Javy (cca 2,5 miliónů řádků kódu) a Java bude dále vyvíjena jako open source.“, viz [7].

#### **Výhody**

- Nezávislost na architektuře – vytvořený systém běží na libovolném operačním systému a libovolné architektuře. Podle konkrétní architektury (platformy) může být přizpůsoben vzhled i chování aplikace.
- Podpora zpracování více-vláknových aplikací.
- Dynamičnost – knihovna může být za chodu rozšiřována o nové třídy a funkce, a to jak z externích zdrojů, tak vlastním programem.

#### **Nevýhody**

- Ve srovnání Javy například s C++ je start programů v Javě pomalejší. Je to způsobeno nutností program nejprve přeložit a potom teprve spustit. Tato nevýhoda může být částečně eliminována využitím mechanismů JIT a HotSpot, tj. často prováděné nebo neefektivní části kódu se přeloží do strojového kódu, čímž se program zrychlí.
- Další nevýhodou je větší paměťová náročnost způsobená zaváděním celého běhového prostředí do paměti.



Obě popsané nevýhody nejsou ve většině případů problémem, protože dnešní výkonnost a hardwarové parametry stolních počítačů daleko převyšují nároky Javy.

#### **4.3.1.1.2. Programovací jazyk Prolog**

Prolog je „logický programovací jazyk. Patří mezi tzv. deklarativní programovací jazyky, ve kterých programátor popisuje pouze cíl výpočtu, přičemž přesný postup, jakým se k výsledku program dostane, je ponechán na libovůli systému. Prolog se snaží o pokud možno abstraktní vyjádření faktů a logických vztahů mezi nimi s potlačením imperativní složky. Prolog je využíván především v oboru umělé inteligence a v počítačové lingvistice (obzvláště zpracování přirozeného jazyka, pro nějž byl původně navržen). Syntaxe jazyka je velice jednoduchá a snadno použitelná právě proto, že byl původně určen pro počítačově nepřiliš gramotné lingvisty. Prolog je založen na predikátové logice prvního řádu; konkrétně se omezuje na Hornovy klauzule. Běh programu je pak představován aplikací dokazovacích technik na zadané klauzule. Základními využívanými přístupy jsou unifikace, rekurze a backtracking.“, viz [6].

#### **SWI-Prolog**

SWI-Prolog je multiplatformní a relativně malý a interpret jazyka Prolog. SWI-Prolog byl vyvinut jako otevřené programové prostředí, poskytující obousměrný interface do jazyka C, což v tomto období nebylo poskytováno žádnou jinou implementací Prologu. Tato implementace odpovídá normě ISO. K SWI-Prologu existují také vývojová prostředí, grafický debugger a také nástroje pro síťovou komunikaci a pro grafické aplikace (toolkit XPCE). SWI-Prolog je licencován pod LGPL (Lesser General Public License) - jádro systému, knihovny jsou licencovány pod GPL (General Public License) s dodatkem, který umožňuje použití i v ne-GPL aplikacích.

#### **JIProlog**

JIProlog (Java Internet Prolog) je multiplatformní interpret prologu pro Javu, který jednoduše propojuje Javu a Prolog. Umožňuje volání predikátů v prologu bez použití nativního kódu, což je obvyklé u jiných interpretů. Kromě volání prologu z Javy umožňuje také volání metod Javy

z prologu v podstatě stejným způsobem. Je použitelný s Javou ve verzi nejméně 1.1. V rámci základní distribuce je dodáváno vývojové prostředí, které zjednodušuje celý vývoj.

JIProlog je poskytován pod Shareware licenci. Jedná se o neexkluzivní nepřenositelnou licenci, která umožňuje uživateli jeho stahování a koncové použití bez jakýchkoliv poplatků pouze pro nekomerční a zákonné účely, viz Seznam příloh na CD.

#### **4.3.1.2. Zpracování dat**

Pro získání dat ze souborů XML a XSD se nabízela podpora integrovaná přímo v interpretu jazyka JIProlog, respektive v jeho modulu pro práci s XML (XML-module). Nicméně tato podpora je zprostředkována s využitím DOM (Data Object Model) rozhraní. Jedná se o objektově orientovanou reprezentaci XML dokumentu, respektive o API umožňující přístup či modifikaci obsahu, struktury nebo stylu dokumentu, či jeho částí. Při použití tohoto rozhraní je celý objektový model při zpracování uložen v paměti a při velikosti dostupných datových souborů (cca 24MB), by paměťové nároky výsledné aplikace byly velké, ale větším problémem je velmi pomalé vyhledávání informací v takto velkých modelech. Toto rozhraní je také podporováno samotným programovacím jazykem Java.

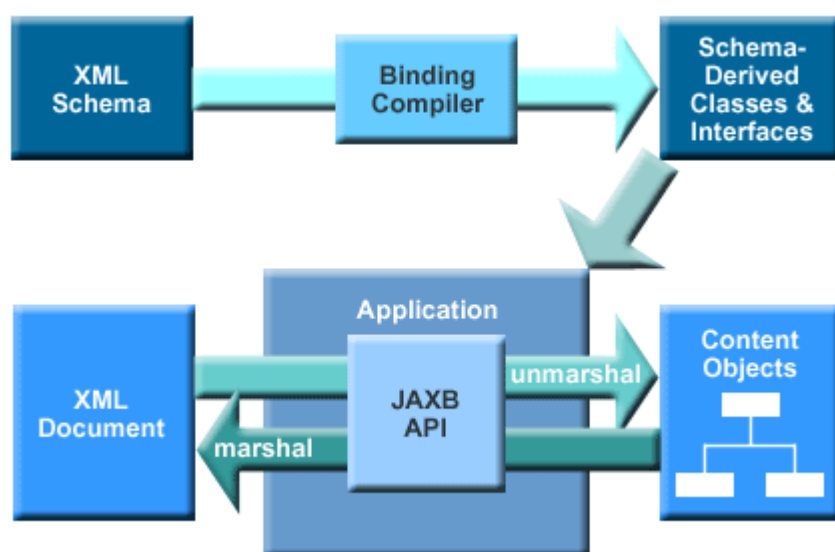
Alternativou DOM rozhraní je použití SAX (Simple Api for XML). Jedná se o událostmi řízené rozhraní, což znamená, že jde o rozhraní nižší úrovně než DOM. Tato rozhraní jsou velmi úsporná na paměť a jsou i časově efektivnější a to hlavně proto, že nedefinuje žádný obraz dat v paměti. Pro načtení informací ze souboru se vytvářejí speciální pravidla pro každou datovou strukturu a její atributy. Při složitosti struktury XML souborů, jak je vidět na obrázku 4.1., by bylo vytvoření pravidel časově velmi náročné a při změně struktury by se musela vždy měnit příslušná část pravidel.

Třetí alternativou je využití technologie JAXB (Java Architecture for XML Processing), viz kapitola 4.3.1.3. Obě již zmíněné technologie pro zpracování XML dokumentů SAX a DOM, jsou poskytovány v rámci společného balíčku JAXP (Java API for XML Processing). Oba přístupy převedou obsah XML dokumentu na objekty v logické struktuře, což zpřístupní obsah dokumentu pro vyvíjenou aplikaci. Výhody i nevýhody obou přístupů byly již popsány v předešlém textu. Oproti tomu JAXB po provedení deserializace může ihned zpřístupnit a zobrazit veškerá data obsažená v XML dokumentu pomocí obsahu vytvořených Java objektů.

Není potřeba vytvářet žádný parser jako v případě DOM nebo psát vlastní handler pro zpracování obsahu jako v případě SAX.

#### 4.3.1.2.1. JAXB

JAXB (Java Architecture for XML Binding) umožňuje mapování XML dokumentů na Java objekty a to oběma směry. Toto mapování musí být řízeno pravidly, která jsou dána XML schématem, které popisuje strukturu daného XML dokumentu. Základní schéma architektury JAXB je vidět na následujícím obrázku, viz Obr. 4.2.



Obr. 4.2. Schéma architektury JAXB, viz Ort [12]

Celé použití spočívá v následujících etapách:

- vytvoření (získání) XML schématu a příslušných XML dokumentů;
- vygenerování Java tříd na základě daného schématu;
- deserializace (unmarshaling) – konverze XML dokumentu do instancí příslušných tříd;
- serializace (marshaling) – konverze instancí tříd do XML reprezentace.

#### 4.3.1.3. Vývojové prostředí

Existuje celá řada vývojových prostředí pro Javu, například Eclipse, Idea, Netbeans atd. Každé z nich má své výhody a nevýhody a většinou se každý rozhodne pro to prostředí, ve kterém je zvyklý pracovat. Pro účely této práce bylo vybráno vývojové prostředí Netbeans verze 6.5. Jedná

se o Open Source projekt Firmy Sun Microsystems založený v červnu 2000. Umožňuje uživateli psát, překládat a ladit aplikace v programovacím jazyce Java - ale stejně může podporovat jakýkoliv jiný programovací jazyk. Existuje také velké množství modulů, které prostředí Netbeans rozšiřují. Jedná se o bezplatně šiřitelný produkt, který je možné používat bez jakýchkoliv omezení.

#### 4.3.2. Návrh uživatelského rozhraní

Celé uživatelské rozhraní by mělo být implementováno v Javě.

Mělo by umožnit uživateli:

- jednoduše a srozumitelně formulovat jeho požadavky na vyhledávání včetně zohlednění preferencí jednotlivých kritérií;
- přehledné zobrazení výsledků a získání podrobných informací o jednotlivých pracovních nabídkách.

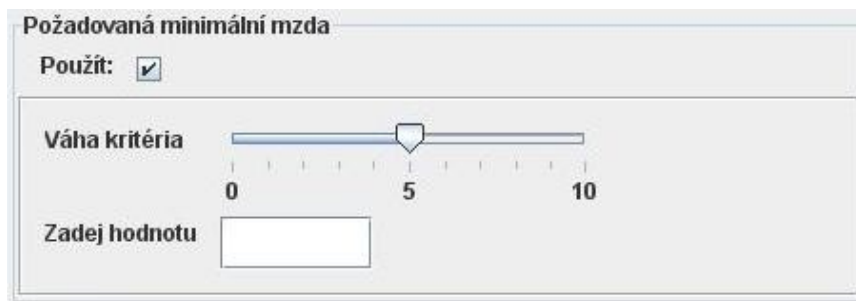
Uživatelské rozhraní bude tedy rozděleno do dvou částí a to:

- část pro specifikaci a konstrukci dotazu;
- část pro zobrazení výsledků.

Navíc by mělo být uživateli umožněno nahrát do systému svá data (svůj XML soubor).

##### 4.3.2.1. Specifikace hodnot a vah kritérií

Prototypový návrh panelu pro specifikaci jednoho kvantitativního kritéria je vidět na následujícím obrázku, viz Obr. 4.3.



Obr. 4.3. Návrh panelu pro specifikaci kvantitativního kritéria

V levém horním rohu je zobrazen název kritéria. Pod názvem je možnost specifikovat, jestli má být dané kritérium při vyhledávání použito. V případě že nebude použito, neměl by mít uživatel možnost manipulovat s vahami, ani hodnotou kritéria. V případě, že kritérium má být použito,

má uživatel možnost specifikovat váhu (významnost) daného kritéria v zadaném rozmezí 0-10, kde 0 znamená nevýznamné kritérium a 10 je hodnota pro nejvýznamnější kritérium.

Po zadání váhy specifikuje uživatel vlastní hodnotu kritéria. V případě kvantitativního kritéria zadá hodnotu do příslušného pole, v případě kvantitativního kritéria budou uživateli nabídnuty možné alternativy, z nichž zvolí jednu nebo více možností.

Běžnou praktikou expertních systémů je, že v rámci dialogu s uživatelem postupně upřesňují informace a požadavky a výsledek je nakonec zobrazen uživateli. V tomto případě by se dalo postupovat podobně. Uživatel by postupně zadával hodnoty a váhy jednotlivých kritérií a nakonec by mu systém zobrazil výsledný seznam pracovních nabídek odpovídající zadaným hodnotám. Nicméně takový postup by byl pro uživatele zcela jistě méně vhodný než možnost zadávat kritéria najednou a vybírat je podle potřeby a po zadání odpovídajících hodnot vyvolat samotné hodnocení a dostat výsledky.

Jednotlivá kritéria by tedy měla být všechna přehledně seřazena v rámci rolovacího okna (okno obsahující posuvné lišty) a uživatel by měl mít možnost zadávat hodnoty podle svého uvážení. Návrh takového rolovacího okna je vidět na obrázku Obr. 4.4.

Kritéria

Požadovaná minimální mzda

Použít: ☒

Váha kritéria

0 5 10

Zadej hodnotu

Obor

Použít: ☒

Váha kritéria

0 5 10

Zadej hodnotu

Administrativa

Okres

Použít: ☒

Váha kritéria

0 5 10

Zadej hodnotu

Jihlava

Vzdělání

Použít: ☒

Váha kritéria

0 5 10

Zadej hodnotu

Nižší střední

Obr. 4.4. Návrh panelu pro seznam kritérií

#### 4.3.2.2. Zobrazení výsledků

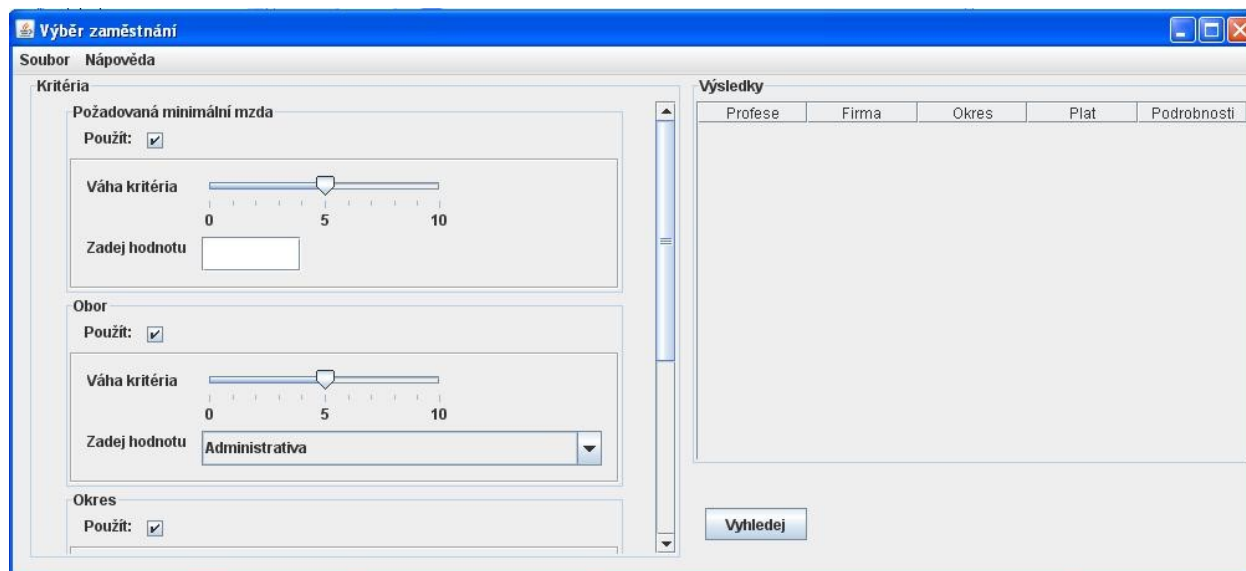
Návrh panelu pro zobrazení výsledků je vidět na obrázku 4.4. Jednotlivé záznamy by měly být seřazeny do tabulky podle výsledného hodnocení. Tabulka by měla obsahovat pouze základní charakteristiku výsledných pracovních nabídek, ale uživateli by mělo být umožněno jednoduchým způsobem získat i podrobné informace například pomocí kliknutí na příslušný odkaz (tlačítko).



Obr. 4.5. Návrh panelu pro zobrazení výsledů

#### 4.3.2.3. Celkový návrh

Zde jsou shrnuty jednotlivé části grafického rozhraní a na obrázku 4.6. je zobrazen celkový návrh uživatelského rozhraní. V levé části je panel zobrazující seznam kritérií, kde uživatel zadává váhy a hodnoty jednotlivých kritérií, v pravé části jsou poté zobrazeny výsledky.



Obr. 4.6. Celkový návrh uživatelského rozhraní

## **4.4. Implementace expertního systému**

V této kapitole bude popsána vlastní implementace výsledného systému. Nejdříve je popsána implementace prototypu, následně jeho hodnocení a na základě tohoto hodnocení finální implementace systému.

### **4.4.1. Prototyp**

Jak je již zmíněno v kapitole popisující vývoj systému prototypováním, měla by první prototypová verze obsahovat pouze základní funkcionalitu výsledného systému, respektive odpovídat hlavním požadavkům na systém a měla by také obsahovat uživatelské grafické rozhraní. Prototyp bude tedy aplikace, která bude napsána v Javě, a grafické uživatelské rozhraní bude odpovídat návrhu, jak byl popsán v kapitole 4.3.2. Bude umožňovat načíst data ze vstupního XML souboru, provést hodnocení a poté zobrazit výsledky uživateli.

#### **4.4.1.1. Zpracování vstupních dat**

Prvotní úkol, který musí být vyřešen, je získání potřebných informací obsažených v XML a XDS souborech dostupných na stránkách Integrovaného portálu Ministerstva práce a sociálních věcí. Pro tyto účely budou použity soubory:

- „*vm20110318.xml*“ – datový soubor formátu XML obsahující údaje o volných pracovních místech celé ČR k datu 19.3.2011.
- „*volnamista-2-2.xsd*“ – soubor ve formátu XSD udávající strukturu datového souboru
- „*ciselniky-1-1.xsd*“ – soubor ve formátu XSD obsahující potřebné číselníky

Pro zpracování těchto souborů byla použita technologie JAXB, viz kapitola 4.3.1.2. V prvním kroku je třeba získat (vytvořit) vstupní data (XML a XSD soubory). Ve druhém kroku je třeba na základě XSD souborů vygenerovat příslušné Java třídy. Vzhledem k tomu, že vývoj prototypu probíhal převážně na systému Mac Os, byl pro vygenerování použit následující příkaz:

```
„xjc.sh -p packageName volnamista.2-2.xsd -d cilovyAdresar“
```

- atribut *-p* specifikuje název balíčku, kde jsou umístěny zdrojové XSD soubory;
- atribut *-d* specifikuje cílový adresář, do kterého se mají vygenerovat výsledné třídy.



Při generování s použitím neupravených vstupních souborů nastal problém. Objevovala se chyba ve vygenerovaných třídách - chyběla třída obsahující číselník pro vzdělání. Bylo to způsobeno stejným pojmenováním elementů v XSD souborech. V souboru „*volnamista-2-2.xsd*“ se vyskytuje element „Vzdelani“ a odpovídající číselník v souboru „*ciselniky-1-1.xsd*“ je pojmenován „VZDELANI“, což způsobí při generování přepsání těchto tříd, protože Java nerozlišuje velikost písmen v názvech jednotlivých tříd. Řešením je jednoduchá úprava souboru s číselníky – přidání přípony k názvu elementu, například „Vzdelanix“ a znovu vygenerování tříd.

Nyní už se ve výsledném výpisu objevovala pouze dvě varování. Nebylo možné vygenerovat některé číselníky kvůli překročení jejich stanovené velikosti. Nicméně tyto číselníky nejsou pro účely této práce potřebné. Obsah vygenerovaného balíčku je vidět na obrázku 4.6.



Obr. 4.6. Třídy vygenerované s použitím JAXB

Nyní je třeba provést konverzi obsahu XML dokumentu do instancí vygenerovaných tříd. Dále je vidět ukázka kódu, který tuto konverzi provádí.

...

```
JAXBContext jc = JAXBContext.newInstance("gen.schema");  
Unmarshaller unmarshaller = jc.createUnmarshaller();  
unmarshaller.setSchema(SchemaFactory.newInstance(XMLConstants.W3C_XML_SCHEMA_NS_URI).newSchema(new File("volnamista-2-2.xsd")));  
VOLNAMISTA volnam = (VOLNAMISTA) unmarshaller.unmarshal(new File("vm20110318.xml"));
```

...

Celý postup začíná vytvořením instance třídy „JAXBContext“, které se jako vstupní parametr konstruktoru zadá název balíčku s vygenerovanými třídami. Dále se vytvoří instance třídy „Unmarshaller“, které se volitelně nastaví pomocí metody „setSchema“ XML schéma pro validaci vstupního XML dokumentu a nakonec se pomocí metody „unmarshal“ provede konverze obsahu XML dokumentu do instancí tříd. V tomto případě vznikne instance třídy VOLNAMISTA, která obsahuje veškeré zpracované informace dostupné pomocí příslušných vnitřních metod.

#### **4.4.1.2. Transformace informací do znalostní báze**

Pro vytvoření znalostní báze je třeba mít funkční interpret jazyka prolog pro Javu. V tomto případě byl použit JIProlog popsáný v kapitole 4.3.1.1.2.

Nejdříve je vytvořena instance třídy „JIPEngine“. Pomocí metody „loadLibrary“ jsou nahrány potřebné knihovny, například „jipxsets“ pro podporu třídících a vyhledávacích funkcí v množinách a seznamech. Dále se pomocí metody „consultFile“ nahrají potřebné zdrojové soubory prologu, například „qsort.pl“ obsahující metodu quicksort pro třídění seznamů, která je použita pro výsledné seřazení výsledků.

Báze faktů je vytvořena ze seznamu volných pracovních míst, který je získán z instance třídy „VOLNAMISTA“ pomocí metody „getVOLNEMISTO“. Postup je znázorněn v následující ukázce kódu:

...

*for (VOLNEMISTO volnemisto : volnaMista) {*

*jip.asserta(jip.getTermParser().parseTerm("volneMisto(atributyProHodnoceni)."));*

*}*

...

V cyklu se prochází seznam volných míst a pro každý jednotlivý záznam je do báze faktů vložen fakt „volneMisto“ s potřebnou sadou atributů, které jsou potřeba pro vlastní hodnocení.

#### 4.4.1.3. Uživatelské rozhraní

Uživatelské rozhraní odpovídá návrhu rozhraní, jak je prezentován v kapitole 4.3.2.

#### 4.4.1.4. Hodnocení variant

Samotné hodnocení variant je závislé na zadaných vahách jednotlivých kritérií a jejich hodnotách. Po odeslání požadavku na výsledné hodnocení jsou příslušné hodnoty kritérií a vah vloženy do systému. Samotná metoda pro výpočet hodnocení variant byla pro účely prototypu nahrazena metodou, která jako vstupní parametr akceptuje volné místo a hodnoty kritérií a jejich vah a jako výsledek vrátí náhodné číslo v intervalu <0-100>. Po výpočtu hodnocení varianty je toto hodnocení vloženo (aktualizováno) jako jeden z atributů faktů „volneMisto“. Další postup je nastíněn v následující ukázce kódu.

...

*JIPTerm queryTerm =*

*jip.getTermParser().parseTerm("findall(Y/X,(volneMisto(X,Y,\_,\_,\_),Y>0),Bag), qsort(Bag, R)");*

*JIPQuery jipQuery = jip.openSynchronousQuery(queryTerm);*

*JIPTerm solution = null;*

*while ((solution = jipQuery.nextSolution()) != null) {*

*...*

*solution.getValue();*

*...*

*}*

...

Nejdříve je formulován dotaz v podobě termu, který najde všechna volná místa, která mají hodnocení větší než nula. Proměnná X odpovídá identifikačnímu číslu záznamu (číslo, které je

unikátní a jednoznačně identifikuje jednotlivé záznamy o pracovních nabídkách ve znalostní databázi). Proměnná Y odpovídá výsledné hodnotě hodnocení. Jako výsledek tedy vznikne seznam dvojic čísel, identifikační číslo a hodnota hodnocení, seřazený podle hodnoty hodnocení.

#### **4.4.1.5.    Prezentace výsledků**

Výsledkem po hodnocení variant je seznam dvojic čísel, identifikační číslo a hodnota hodnocení, seřazený podle hodnoty hodnocení. Tento seznam neobsahuje žádné další informace. Další informace nejsou třeba. Fakta uložená ve znalostní databázi obsahují pouze informace nutné k hodnocení variant, ale ne podrobné informace k jednotlivým nabídkám jako například slovní popis nabídky, firmu, která inzerát zadala, atd.

Všechny tyto informace by měly být zobrazeny uživateli. Proto se nyní z výsledného seznamu vezme z prvních pozic počet záznamů, který má být zobrazen uživateli. Postupně se procházejí všechny tyto záznamy a podle identifikačního čísla se dohledají záznamy obsažené v seznamu volných míst vzniklých při zpracování vstupních XML souborů. Poté jsou tyto záznamy vhodně zobrazeny v tabulce výsledků.

#### **4.4.2.       Hodnocení prototypu a návrhy vylepšení**

Prototyp v podstatě odpovídá všem základní požadavkům na systém. Jediným problémem je značná časová a paměťová náročnost zadávání informací do báze faktů, jejich následná aktualizace (doplnění hodnoty výsledného hodnocení) a konečné seřazení záznamů do výsledného seznamu.

Vstupní XML dokument obsahuje přibližně 12500 pracovních nabídek, tím ve znalostní databázi vzniká stejný počet faktů. Tato operace (vytvoření znalostní báze) trvá přibližně 30s. Výpočet výsledků a aktualizace hodnot trvá přibližně stejně dlouho a finální seřazení trvá přibližně 40s. Navíc kvůli velkému počtu rekurzivního zanoření při finálním seřazení je nutné zvýšit velikost paměti zásobníku pro Javu na 10MB z původních 512KB, aby program fungoval správně. Tyto hodnoty byly měřeny na notebooku s procesorem Intel 1,6GHz, 760MB RAM.

Stejný průběh operací implementovaný přímo v Javě zabere několik sekund. Odpadá nutnost vkládání faktů do báze dat a samotné hodnocení a seřazení výsledků proběhne během několika sekund. Z uživatelského hlediska je tedy vhodnější ve finální implementaci implementovat celý

systém přímo v Javě. Tento návrh je podpořen i faktem, že část implementovaná v prologu je pro tento typ problému ve své podstatě nadbytečná.

#### **4.4.3. Implementace výsledného systému**

Vzhledem k hodnocení prototypu je z uživatelského hlediska nutné časově zefektivnit celý proces hodnocení variant, což může být provedeno například:

- nahrazením použitého interpretu jazyka prolog efektivnějším interpretem;
- nahrazením funkcionality implementované v prologu efektivnější interpretací v Javě.

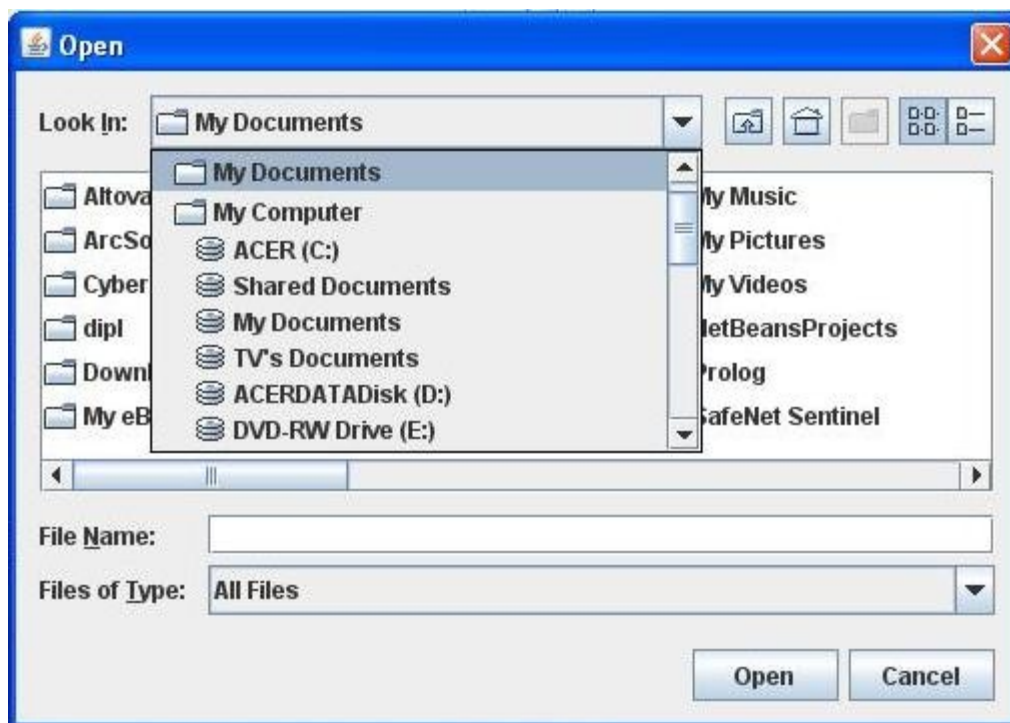
V rámci možnosti nahrazení interpreta jazyka prolog by přicházelo v úvahu použití pravidlového systému založeného na Javě – Jess. Jedná se o pravidlový systém a skriptovací prostředí napsaná v Javě vyvinuté Ernestem Friedman-Heillem. Použití Jess umožňuje vybudovat software v Javě umožňující provádět odvozování s použitím znalostí, které jsou zadávány v deklarativní formě. Jess je malý, lehký a jeden z nejrychlejších dostupných pravidlových systémů pro Javu. Jess obsahuje kompaktní vývojářské prostředí založené na platformě Eclipse.

Druhou alternativou je nahrazení funkcionality implementované s využitím JIPrologu implementací v Javě.

Pokud by byla realizována první varianta, znamenalo by to v podstatě začít vývoj od začátku, tj. vytvořit nový prototyp a na základě výsledků hodnocení postupovat dále. Z uživatelského hlediska i časové efektivity aplikace je vhodnější druhá varianta, a proto bude také použita ve výsledné aplikaci.

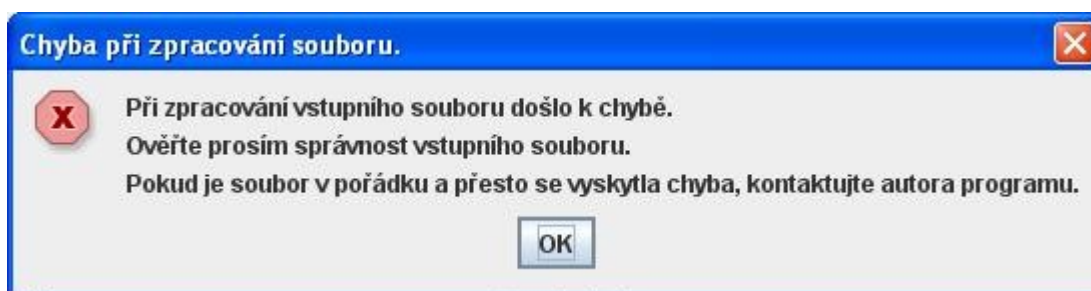
##### **4.4.3.1. Zpracování vstupních dat**

Zpracování vstupních dat zůstává stejné jako u prototypové verze, tj. načtení vstupního souboru do objektů v Javě s použitím technologie JAXB. Pro načtení vstupního souboru do systému byl přidán dialog pro otevírání souborů viz Obr. 4.7. dostupný přes menu aplikace.



Obr. 4.7. Dialog pro načtení vstupního XML souboru

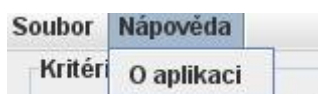
Po otevření příslušného souboru se načtou vstupní data do systému. Pokud se při zpracování souboru vyskytne chyba, je uživatel informován zobrazením vhodné zprávy, například Obr. 4.8.



Obr. 4.8. Chybová zpráva při zpracování vstupního souboru

#### 4.4.3.2. Uživatelské rozhraní

Základní uživatelské rozhraní v zásadě odpovídá prototypové verzi, ale bylo dále rozšířeno. Do aplikace bylo přidáno menu (Obr 4.9.), které umožňuje uživateli nahrát vstupní XML soubor, viz předchozí kapitola, a zobrazit informace o aplikaci (Obr. 4.10.)



Obr. 4.9. Menu aplikace



Obr. 4.10. Informace o aplikaci

Dalším požadavkem na výslednou aplikaci bylo zobrazení podrobných informací o výsledných nabídkách práce. V tabulce výsledků (Obr. 4.11) jsou uvedeny základní charakteristiky jednotlivých nabídek práce.

Profese	Firma	Okres	Plat	Podrobnosti
Advokáti	Vejmelka & W...	Hlavní město ...	50000	Podrob...
Advokátní kon...	JUDr. Miroslav...	Šumperk	13000	Podrob...
Advokátní kon...	M.B.A. Financ...	Jeseník	10000	Podrob...
Advokátní kon...	Mgr. Zdeněk P...	Děčín	13000	Podrob...
Advokátní kon...	Michael Kis M...	Chomutov	12000	Podrob...
Advokátní kon...	Vejmelka & W...	Hlavní město ...	25000	Podrob...
Agenti dopravy...	Charles Bridg...	Ostrava-město	18000	Podrob...
Agenti dopravy...	Panasonic Aut...	Pardubice	18000	Podrob...
Agenti dopravy...	PSOTA transp...	Pardubice	25000	Podrob...
Agenti dopravy...	Duke Manufac...	Praha-východ	35000	Podrob...
Agenti dopravy...	Magna Cartec...	Jindřichův Hra...	20000	Podrob...
Agenti dopravy...	DACHSER E....	Brno-venkov	20000	Podrob...
Analytici a vývo...	HELLA AUTO...	Šumperk	20000	Podrob...
Analytici a vývo...	COPY GENER...	Hlavní město ...	20000	Podrob...
Aranžéři	SPAR Česká ...	Frýdek-Místek	9500	Podrob...
Aranžéři	KARAFa, spol...	Most	10500	Podrob...
Architekti, spe...	ZF Engineerin...	Plzeň-město	23000	Podrob...
Architekti, spe...	Haniš srubov...	Hradec Králové	13000	Podrob...
Architekti, spe...	IDIADA CZ a.s.	Hradec Králové	20000	Podrob...
Asistenti na vy...	Západočeská ...	Plzeň-město	13000	Podrob...
Asistenti na vy...	Západočeská ...	Plzeň-město	13000	Podrob...
Asistenti na vy...	Západočeská ...	Plzeň-město	13000	Podrob...

Obr. 4.11. Přehled výsledných nabídek práce

U každé této nabídky je v posledním sloupci tlačítko „Podrobnosti“. Po kliknutí na toto tlačítko jsou uživateli zobrazeny podrobné informace k příslušné nabídce práce, viz Obr 4.12.

<b>Základní údaje</b> Firma : Vejmelka & Wünsch, v.o.s. Profese : Advokáti Vzdělání : Vysokoškolské Mzda (min) : 50000	<b>Adresa Pracoviště</b> Okres : Hlavní město Praha Obec : Praha Ulice : Italská 753 PSČ : 120 00
<b>Kontaktní osoba</b> Miluška Vrzáčková telefon : +420 737 797 000 email : miluska.vrzackova@vejwun.cz	
<b>Podrobnosti</b> Advokátní kancelář se zahraniční klientelou přijme advokáta s výbornou znalostí platného práva a anglického, event. německého jazyka, se zaměřením na obchodní právo. Nabízíme příjemné prostředí, odpovídající zajímavé finanční ohodnocení. Svůj životopis v českém a anglickém jazyce zašlete prosím k rukám PhDr. Milušky Vrzáčkové, e-mail miluska.vrzackova@vejwun.cz.	
<b>WWW</b> <a href="http://www.vejmelkawuenssch.cz">http://www.vejmelkawuenssch.cz</a>	

Obr. 4.12. Ukázka formuláře podrobných informací práce nabídky

#### 4.4.3.3. Hodnocení variant

Samotné hodnocení variant je závislé na zadaných vahách jednotlivých kritérií a jejich hodnotách. Po odeslání požadavku na výsledné hodnocení jsou příslušné hodnoty kritérií a vah přečteny z grafického rozhraní a poté vloženy do systému.

Pro celkové hodnocení variant byla v našem případě použita metoda přímého stanovení vah kritérií a následná aplikace metody lineárních dílčích funkcí užitku na jednotlivé varianty.

#### Váhy kritérií

Nejdříve jsou stanoveny normované váhy jednotlivých kritérií. Protože zadaná stupnice vah kritérií  $<0,10>$  obsahuje hodnotu „0“, je ke každé zadané hodnotě nejdříve přičtena 1, aby



nedocházelo k dělení nulou. Normování vah odpovídá dělení vah jednotlivých kritérií součtem vah pro všechna kritéria, což odpovídá vzorci 3.1.

### **Hodnocení variant dle jednotlivých kritérií**

Následuje hodnocení jednotlivých variant metodou lineárních dílčích funkcí užitku. Tato metoda se provádí pro každý typ kritérií (kvalitativní a kvantitativní) jiným způsobem.

Vztah pro výpočet hodnocení variant podle kvantitativních kritérií je dán vzorcem 3.5. V našem případě se jedná o kritérium minimální mzda. Uživatel zadá preferovanou hodnotu minimální mzdy a odpovídající váhu kritéria. Nejdříve se v celém souboru variant určí pro dané kritérium maximum a minimum. Výsledné hodnocení se pak počítá podle vzorce 3.5. a výsledek hodnocení je nakonec vynásoben vahou kritéria. Pro kritérium minimální mzda bylo určení hodnocení ztíženo variabilitou dat. Většina hodnot je dána jako měsíční mzda, ale u několika variant je hodnota zadána jako hodinová mzda. V těchto případech bylo použito vynásobení hodinové mzdy počtem hodin v měsíci, tj. 160, aby byly hodnoty srovnatelné.

Postup pro výpočet hodnocení variant podle kvalitativních kritérií je dán expertním stanovením dílčích ohodnocení. V našem případě se jedná o kritéria okres, obor a minimální požadované vzdělání. Uživatel nejprve vybere z možných hodnot kritéria požadované hodnoty a nastaví příslušné váhy. Pro jednotlivé varianty se poté nastaví užitek na:

- 0 v případě, že zadaná hodnota neodpovídá příslušným hodnotám kritéria dané varianty;
- 1 v případě, že zadaná hodnota odpovídá příslušným hodnotám kritéria dané varianty.

Výsledek hodnocení je nakonec vynásoben vahou kritéria. Jednotlivé varianty neobsahovaly vždy pouze jednu hodnotu daného kritéria, ale seznam hodnot, což komplikovalo hodnocení. Řešením bylo neporovnávat hodnoty přímo, ale projít daný seznam a zjistit jestli obsahuje zadanou hodnotu.

Pokud se při hodnocení objevila varianta, která neměla pro dané kritérium zadanou hodnotu, bylo dílčí hodnocení užitku této varianty vzhledem k danému kritériu stanoveno na hodnotu 0. Celkové hodnocení variant odpovídá součtu dílčích hodnocení variant podle jednotlivých kritérií.

#### **4.4.3.4.   Prezentace výsledků**

Jednotlivé varianty jsou seřazeny sestupně, podle hodnot výsledného hodnocení a poté jsou zobrazeny uživateli v tabulce, respektive podrobných výpisech, jak je uvedeno v kapitole 4.4.3.2.

#### **4.4.3.5.   Použití výsledné aplikace**

Celá aplikace je implementována v Javě. Výstupem je spustitelný „*gui.jar*“ soubor, který je možné nalézt na přiloženém CD, viz Seznam příloh na CD.

Předpokladem ke spuštění aplikace je nainstalovaný interpret jazyka Java verze 1.5 nebo vyšší. Aplikaci je možné spustit příkazem „*java -jar cesta/gui.jar*“, kde část „cesta“ je nahrazena skutečnou cestou k souboru „*gui.jar*“ na přiloženém CD.

## Shrnutí a závěr

Na základě analýzy současného stavu Integrovaného portálu Ministerstva práce a sociálních věcí byla v rámci této práce vytvořena aplikace umožňující uživatelům vyhledávání pracovních nabídek nejenom podle zadaných hodnot kritérií, ale také podle zadaných vah odpovídajících preferencím uživatelům.

V první fázi byla vytvořena metodika takového typu vyhledávání informací, kdy jsou jednotlivé varianty hodnoceny na základě vah stanovených metodou přímého stanovení vah kritérií a samotného hodnocení variant metodou lineárních dílčích funkcí užítku.

Následně byl na základě tohoto návrhu vytvořen prototyp hybridního expertního systému implementovaný v Javě s využitím interpretu jazyka prolog pro Javu JIProlog. Součástí tohoto prototypu bylo i grafické uživatelské rozhraní.

Po hodnocení prototypu bylo nutné zefektivnit proces hodnocení variant a zobrazení výsledků z hlediska časové náročnosti. Po zvážení alternativ byla ve finální aplikaci nahrazena implementace hodnocení napsaná v prologu implementací v Javě. Výsledná aplikace odpovídá požadavkům vniklých na základě SWOT analýzy současného stavu, což znamená, že je přenositelná mezi platformami.

Samotné výsledky poskytnuté výslednou aplikací usnadňující výběr pracovních míst pro uchazeče o práci na pracovních úřadech a zohledňují při vyhledávání preference uživatelů. Tím přispívají ke zpřesnění výběru vhodných nabídek.

Některé možnosti dalšího vývoje jsou nastíněny v provedené SWOT analýze, tj. tato aplikace by mohla být v budoucnu použita online s využitím technologie Java Web Start. Vzhledem k současné architektuře systému by toto rozšíření nemělo být problematické. Další rozšíření funkcionality vyplyne z reálného používání této aplikace. Mohlo by se jednat o rozšíření počtu vyhledávacích kritérií, stránkování zobrazených výsledků atd.

## Literatura

[1] Manažerský informační systém. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, , last modified on 9. 2. 2011 [cit. 2011-03-13]. Dostupné z WWW:

<[http://cs.wikipedia.org/wiki/Mana%C5%B8ersk%C3%BD\\_informa%C4%8Dn%C3%AD\\_syst%C3%A9m](http://cs.wikipedia.org/wiki/Mana%C5%B8ersk%C3%BD_informa%C4%8Dn%C3%AD_syst%C3%A9m)>.

[2] Systémy pro podporu rozhodování. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, , last modified on 12. 9. 2010 [cit. 2011-03-13]. Dostupné z WWW:

<[http://cs.wikipedia.org/wiki/Syst%C3%A9my\\_pro\\_podporu\\_rozhodov%C3%A1n%C3%AD](http://cs.wikipedia.org/wiki/Syst%C3%A9my_pro_podporu_rozhodov%C3%A1n%C3%AD)>.

[3] FOTR, J.; Švecová, L.; a kol. Manažerské rozhodování, postupy, metody a nástroje. Praha: Ekopres, s.r.o., 2010. 474 s. ISBN 978-80-86929-59-0

[4] VEBER, J. a kol. Management. Zaklady, prosperita, globalizace. Praha : Management Press, 2000. 700 s. ISBN 80-7261-029-5.

[5] MERRITT, Denis; Building Expert Systems in prolog; Amzi! inc.; Lebanon: 5861 Greentree Road; 2000. ISBN 978-03-87970-16-5

[6] Prolog (programovací jazyk). In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, , last modified on 28. 1. 2011 [cit. 2011-03-17]. Dostupné z WWW: <[http://cs.wikipedia.org/wiki/Prolog\\_%28programovac%C3%AD\\_jazyk%29](http://cs.wikipedia.org/wiki/Prolog_%28programovac%C3%AD_jazyk%29)>

[7] Java (programovací jazyk). In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, , last modified on 13.3.2011 [cit. 2011-03-21]. Dostupné z WWW: <[http://cs.wikipedia.org/wiki/Java\\_%28programovac%C3%AD\\_jazyk%29](http://cs.wikipedia.org/wiki/Java_%28programovac%C3%AD_jazyk%29)>.

[8] ZENDULKA, J.; Projektování programových systémů - 2 Životní cyklus programového díla, [cit. 2011-03-21]. Dostupné z WWW:

[http://www.fit.vutbr.cz/study/courses/PPS/public/pdf/2\\_Ziv\\_cykl.pdf](http://www.fit.vutbr.cz/study/courses/PPS/public/pdf/2_Ziv_cykl.pdf)

[9] SOCHOR, J.; Analýza a návrh systémů, [cit. 2011-03-21]. Dostupné z WWW:

<http://www.fi.muni.cz/~sochor/PB007/Slajdy/UvodP007.pdf>

[10] Extensible Markup Language. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, last modified on 23.2.2011 [cit. 2011-04-06]. Dostupné z WWW: <[http://cs.wikipedia.org/wiki/Extensible\\_Markup\\_Language](http://cs.wikipedia.org/wiki/Extensible_Markup_Language)>.

[11] XML Schema. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, last modified on 22.3.2011 [cit. 2011-04-06]. Dostupné z WWW: <[http://cs.wikipedia.org/wiki/XML\\_Schema](http://cs.wikipedia.org/wiki/XML_Schema)>.

[12] ORT, E; MEHTA, B; Java Architecture for XML Binding (JAXB) [online]. 2003, [cit. 2011-04-06]. Dostupný z WWW: <<http://www.oracle.com/technetwork/articles/javase/index-140168.html>>.

## Seznam zkratek

<b>CD</b>	Compact Disk
<b>DOM</b>	Data Object Model
<b>DSS</b>	Decision Support System
<b>ES</b>	Expert System
<b>GPL</b>	General Public Licence
<b>JAXB</b>	Java Architecture for XML Processing
<b>JAXP</b>	Java API for XML Processing
<b>JIProlog</b>	Java Internet Prolog
<b>JNI</b>	Java Native Interface
<b>LGPL</b>	Lesser General Public Licence
<b>MIS</b>	Management Information System
<b>NUTS</b>	Nomenklatura územních statistických jednotek
<b>SAX</b>	Simple Api for XML
<b>SWOT</b>	Strengths, Weaknesses, Opportunities, and Threats
<b>XML</b>	Extensible Markup Language
<b>XSD</b>	XML Schema Definition

## Prohlášení o využití výsledků diplomové práce

Prohlašuji, že

- jsem byl(a) seznámen(a) s tím, že na mou diplomovou (bakalářskou) práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo;
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně, ke své vnitřní potřebě, diplomovou (bakalářskou) práci užít (§ 35 odst. 3);
- souhlasím s tím, že diplomová (bakalářská) práce bude v elektronické podobě archivována v Ústřední knihovně VŠB-TUO a jeden výtisk bude uložen u vedoucího diplomové (bakalářské) práce. Souhlasím s tím, že bibliografické údaje o diplomové (bakalářské) práci budou zveřejněny v informačním systému VŠB-TUO;
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona;
- bylo sjednáno, že užít své dílo, diplomovou (bakalářskou) práci, nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Ostravě dne .....

.....

jméno a příjmení studenta

Adresa trvalého pobytu studenta:

.....

## Seznam příloh na CD

Součástí této práce je i přiložené CD, které obsahuje:

- zdrojové kódy aplikace ve formě Netbeans projektu (adresář „zdrojove\_kody“);
- vytvořenou aplikaci ve formě spustitelného JAR souboru (adresář „aplikace“);
- obrázky zobrazující strukturu vstupních XML souborů (adresář „obrazky“);
- ukázková data:
  - vstupní XML soubor (adresář „data/xml“);
  - odpovídající XSD šablony (adresář „data/xsd“);
- licenční podmínky pro JIProlog (adresář „JIProlog\_licence“).